UDR-TR-79-72

DEVELOPMENT AND EVALUATION OF L/ESS DATA COMPRESSION TECHNIQUES

Ronald K. Newman
Ronald M. Janning

University of Dayton
Research Institute
Dayton, Ohio

August 1979

Contract Number F33615-78-C-5208

Prepared for:
Aeronautical Systems Division (ASD/ENFSL)
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development and Evaluation of L/ESS Data Compression Techniques		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER UDR-TR-79-72
7. AUTHOR(s) Ronald K. Newman, Ronald M. Janning		8. CONTRACT OR GRANT NUMBER(s) F33615-78-C-5208
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Dayton Research Institute, 300 College Park Avenue, Dayton, Ohio 45469		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Aeronautical Systems Division (ASD/ENFSL), Air Force Systems Command, Wright-Patterson Air Force Base, OH 45433		12. REPORT DATE August 1979
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the programs developed for the evaluation of two different methods of compressing the data from the loads/environmental spectra survey (L/ESS) program on three different aircraft. The two different compression techniques involve: (a) saving sections of data when any parameter exceeds a given threshold, and (b) saving the value of an individual parameter when it changes by more than a given window value.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Results of the application of these two compression techniques applied to data recorded on the A-10, C-130, and C-141 aircraft are presented. Complete documentation of the programs is provided for the user and for the programmer who might wish to use the methods on these aircraft or other aircraft.

FOREWORD

This report was prepared by the Aerospace Mechanics Section of the University of Dayton Research Institute, Dayton, Ohio, under Air Force Contract NO. F33615-78-C-5208. The work was initiated and monitored by the Loads Branch, Structures Division, Air Frame Directorate, Deputy for Engineering, Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio with Mr. Steven Guilfoos and Mr. Cyril Peckham as technical monitors.

The project was conducted under the general direction of Dale H. Whitford, Supervisor, Aerospace Mechanics Section with Mr. Ronald K. Newman as Project Engineer. Ronald M. Janning provided the programming effort and assisted in documentation.

TABLE OF CONTENTS

Section		Page
1	INTRODUCTION	1
2	PROGRAM INPUT	2
	2.1 R/T TAPE FORMAT	2
	2.2 CARD INPUT	14
	2.2.1 Input to COMPA1	14
	2.2.2 Input to COMPA2	15
	2.2.3 Input to COMPB1	15
	2.2.4 Input to COMPB2	16
3	DESCRIPTION OF PROGRAMS	17
	3.1 METHOD A	18
	3.1.1 COMPA1 Program	20
	3.1.2 COMPA2 Program	20
	3.2 METHOD B	20
	3.2.1 Program COMPB1	21
	3.2.2 Program COMPB2	23
	3.3 SUBROUTINE PACK	23
	3.4 SUBROUTINE UNPACK	24
4	PROGRAM OUTPUT	25
5	RESULTS	53
6	CONCLUSIONS AND RECOMMENDATIONS	63
APPENDIX		
A	DATA COMPRESSION PROGRAM LISTINGS	64

LIST OF ILLUSTRATIONS

Figure		Page
2.1	BIT and CAL Cycles	12
4.1	Method A Output	27
4.2	Method B Output	30
A.1	Flowchart of Method A	120
A.2	Flowchart of Method B	128

LIST OF TABLES

Table		Page
2.1	General Description of R/T Output Tape	3
2.2	ECU-68/A Multiplexing Format A-10	4
2.3	ECU-67/A Multiplexing Format C-141A/C-130	6
2.4	Documentary Data Encoder	13
3.1	Sampling Rates of Parameters for C-130/C-141	22
4.1	COMPA1 Tape Output Format	33
4.2	COMPA2 Tape Output Format	38
4.3	COMPB1 Tape Output Format Record Type 1	43
4.4	COMPB2 Tape Output Format Record Type 1	48
5.1	Active Data by Parameter Using Method A (A-10)	54
5.2	Active Data by Parameter Using Method A (C-141)	55
5.3	Active Data by Parameter Using Method A (C-130)	56
5.4	A-10 Method B Samples of Active Data	57
5.5	C-141 Method B Samples of Active Data	58
5.6	Comparison of Methods for C-141 Data	60
5.7	Comparison of Methods for A-10 Data	62
A.1	Program COMPA1	66
A.2	Program COMPA2	78
A.3	Program COMPB1	90
A.4	Program COMPB2	101
A.5	Subroutine Pack/Unpack	113
A.6	Programs COMPA1 and COMPA2 Definition of Variables	115
A.7	Program COMPB1 and COMPB2 Definition of Variables	118

SECTION 1
INTRODUCTION

Four separate computer programs were written for the CYBER 175 computer at Wright-Patterson Air Force Base for use in evaluating two different data compression techniques on two different aircraft types. Two different programs were written for the two aircraft types utilizing the same compression technique in order to simplify programming and provide more efficient programs.

The input for the data compression is the Reformatter/Transcriber (R/T) tape generated at Oklahoma City Air Logistics Center by reformatting and transcribing the data recorded on the MXU-553/A Airborne Signal Recorder. The first compression method evaluated (Method A) saves all recorded data during a second when any of the selected parameters examined deviates by more than a specified value from its prescribed mean or when any flight condition changes by more than a specified value from the previous stored value.

The second compression method (Method B) saves values of each individual parameter when it changes by more than a specified value from the previous stored value of that parameter. The maximum(Min) excursion upon change of direction of the parameter is also saved as a "peak". The time of occurrence of these window crossings and peaks is stored with the value.

Comparison of the two compression methods applied to A-10, C-130 and C-141 data are presented in Section 5. A real test of the degree of compression attained with the two methods can be obtained from the sample problem data, for the A-10, presented in Figures 4.1 and 4.2. The total amount of data on the RT tape was 52,810 seconds (102.6 x 10^6 bits). Using Method A, there were 28,657 records (43.0 x 10^6 bits) written to the output tape for a compression ratio of 2.36. With Method B, there were 3554 records (3.82 x 10^6 bits) written to the output tape for a compression ratio of 26.9. These results were obtained with the window values in Method B equal to the threshold values used in Method A where applicable. Representative window values were chosen for all other parameters

examined in Method B that are not analyzed in Method A. Ground data were processed in both of these runs. The Central Processor times used on these runs were 220 seconds and 360 seconds for Method A and Method B, respectively.

One additional run of Method A and Method B was made on the A-10 data, not processing ground data, at the request of the Project Engineer. For Method A thresholds of 23, 13, 8, 13, 13, 4, 4, and 4 were used for NZ, NY, P, Q, R, ALT, A/S and FCTR, respectively. For Method B windows of 7, 6, 10, 9, 9, 12, 14, 14, 4, 1, 1, 8, 4, and 4 were used for NY, NZ, P, Q, R, PDOT, QDOT, RDOT, RCTR, PLA, DSB, SIGMA, ALT, and A/S, respectively. The compression ratios of Method A and Method B were 3.51 and 11.52, respectively.

SECTION 2
PROGRAM INPUT

The input to the data compression programs comes from two sources. Program control is provided by card input and the data to be compressed comes from tape. The tape is that produced by the R/T located at Oklahoma City Air Logistics Center.

2.1 R/T TAPE FORMAT

The output tape from the R/T, the input tape for the computer, contains the inflight recorded data plus some supplementary or header data entered via the ASR33 teletype unit which is part of the R/T. The R/T has the capability of writing either a 7-track or a 9-track tape. Only the 9-track format will be discussed here.

The data on the 9-track cartridge tape is written in physical records of 479 8-bit bytes and the physical records are separated by a one-byte gap. The R/T adds seven 8-bit samples of information to each 479-sample physical record of data, making a set of 486 8-bit samples of data. The R/T groups five of these 486-sample data sets together and writes them on the output tape as a single physical record containing 2430 data samples.

Each set of 486 samples is referred to as a "frame" of data. There are two seconds worth of data in each frame. The first 239 samples are for the first second. The next 240 samples are for the following second of data. The final seven samples of data, supplied by the R/T, contain information regarding the quality of the "frame" of data as it was recorded on the tape cartridge, and are discussed in Table 2.1. Thus, a physical record containing 5 frames contains 10 seconds of data.

The content of each of the 479 samples of data contained in a frame of the R/T output tape for the A-10 is shown in Tables 2.2. The content of the tape for the C-130 and C-141A aircraft is shown in Table 2.3. The two sample numbers refer to the two seconds of data contained in each frame. The following should be

TABLE 2.1

GENERAL DESCRIPTION OF R/T OUTPUT TAPE

Byte or Sample	Description
1-479	Each byte corresponds to a respective multiplexer recording format (Table 2.2). That is, byte 1 is multiplexer sample 2, byte 2 is sample 3,..., byte 479 is sample 480. (Note that sample 1 for the multiplexer is really a one-byte cartridge tape gap which is not transcribed or represented on the output tape.)
480	This byte or sample represents two pieces of information. The four least significant bits are used to designate whether a normal length record was read from the recorder cartridge. When a normal length record is read, an integer nine is represented in these bits. Any other bit pattern indicates a short frame, hence the data in the frame may be invalid. The four most significant bits of this byte are used to indicate an end-of-flight, or more realistically a beginning of flight, since these bits are all on for a minimum of six frames and a maximum of ten frames after the beginning of a flight.
481-482	Value from 1 to 479 which indicates the last correct input byte in those frames in which parity errors have occurred. These two bytes together must be interpreted as a 16-bit unsigned binary integer. (If there are no parity errors the value is set to 482.)
483-484	Number of parity errors encountered by the playback hardware in reading this frame of data from the recorder cartridge. It is also interpreted as a 16-bit unsigned integer.
485-486	If all bits are ones, this is a padding frame which follows the final data frame. At least one such padding frame will always be written on the output tape. More than one will be written if required to complete the last physical record on the tape. If all bits are <u>not</u> ones, then this 16-bit unsigned integer contains the number of this frame as counted from the beginning of the tape.

TABLE 2.2

ECU-68/A MULTIPLEXING FORMAT A-10

Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter
1&241	tape gap & T	31&271	q	61&301	r	91&331	N_y
2&242	V_i	32&272	N_z	62&302	p	92&332	σ
3&243		33&273		63&303	q	93&333	r
4&244	PLA	34&274		64&304	N_z	94&334	p
5&245		35&275		65&305	DD1	95&335	q
6&246	δ_{SB}	36&276	PLA	66&306	DD2	96&336	N_z
7&247	FCTR	37&277		67&307	DD3	97&337	DD7
8&248		38&278	δ_{SB}	68&308	PLA	98&338	DD8
9&249		39&279	FCTR	69&309		99&339	h_p
10&250		40&280		70&310	δ_{SB}	100&340	PLA
11&251	N_y	41&281		71&311	FCTR	101&341	
12&252	σ	42&282		72&312		102&342	δ_{SB}
13&253	r	43&283	N_y	73&313		103&343	FCTR
14&254	p	44&284	σ	74&314		104&344	
15&255	q	45&285	r	75&315	N_y	105&345	
16&256	N_z	46&286	p	76&316	σ	106&346	
17&257	E_1-E_2	47&287	q	77&317	r	107&347	N_y
18&258	E_5-E_9	48&288	N_z	78&318	p	108&348	σ
19&259		49&289		79&319	q	109&349	r
20&260	PLA	50&290		80&320	N_z	110&350	p
21&261		51&291		81&321	DD4	111&351	q
22&262	δ_{SB}	52&292	PLA	82&322	DD5	112&352	N_z
23&263	FCTR	53&293		83&323	DD6	113&353	DD9
24&264		54&294	δ_{SB}	84&324	PLA	114&354	DD10
25&265		55&295	FCTR	85&325		115&355	DD11
26&266		56&296		86&326	δ_{SB}	116&356	PLA
27&267	N_y	57&297		87&327	FCTR	117&357	
28&268	σ	58&298		88&328		118&358	δ_{SB}
29&269	r	59&299	N_y	89&329		119&359	FCTR
30&270	p	60&300	σ	90&330		120&360	

TABLE 2.2 (Concluded)

ECU-68/A MULTIPLEXING FORMAT A-10

Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter
121&361		151&391	FCTR	181&421		211&451	
122&362		152&392		182&422	δ_{SB}	212&452	PLA
123&363	N_y	153&393		183&423	FCTR	213&453	
124&364	σ	154&394		184&424		214&454	δ_{SB}
125&365	r	155&395	N_y	185&425		215&455	FCTR
126&366	p	156&396	σ	186&426		216&456	
127&367	q	157&397	r	187&427	N_y	217&457	
128&368	N_z	158&398	p	188&428	σ	218&458	
129&369	DD12	159&399	q	189&429	r	219&459	N_y
130&370		160&400	N_z	190&430	p	220&460	σ
131&371		161&401		191&431	q	221&461	r
132&372	PLA	162&402		192&432	N_z	222&462	p
133&373		163&403		193&433		223&463	q
134&374	δ_{SB}	164&404	PLA	194&434		224&464	N_z
135&375	FCTR	165&405		195&435		225&465	
136&376		166&406	δ_{SB}	196&436	PLA	226&466	
137&377		167&407	FCTR	197&437		227&467	
138&378		168&408		198&438	δ_{SB}	228&468	PLA
139&379	N_y	169&409		199&439	FCTR	229&469	
140&380	σ	170&410		200&440		230&470	δ_{SB}
141&381	r	171&411	N_y	201&441		231&471	FCTR
142&382	p	172&412	σ	202&442		232&472	
143&383	q	173&413	r	203&443	N_y	233&473	
144&384	N_z	174&414	p	204&444	σ	234&474	
145&385		175&415	q	205&445	r	235&475	N_y
146&386		176&416	N_z	206&446	p	236&476	σ
147&387		177&417		207&447	q	237&477	r
148&388	PLA	178&418		208&448	N_z	238&478	p
149&389		179&419		209&449		239&479	q
150&390	δ_{SB}	180&420	PLA	210&450		240&480	N_z

TABLE 2.3

ECU-67/A MULTIPLEXING FORMAT C-141A/C-130

NOTE: C-130 has same format with S1 instead of NY

Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter
1	GAP	21	NY/S1	41	PS	61	NY/S1	81	DELR	101	NY/S1
2	PITCH	22	STR4	42	STR2	62	PITCH	82	STR4	102	STR2
3	YAW	23	STR5	43	STR3	63	YAW	83	STR5	103	STR3
4	NZ	24	DELF	44	NZ	64	DELE	84	NZ	104	STR6
5	NY/S1	25	DELNG	45	NY/S1	65	E1-4	85	NY/S1	105	DELR
6	STR2	26	PITCH	46	STR4	66	STR2	86	PITCH	106	STR4
7	STR3	27	YAW	47	STR5	67	STR3	87	YAW	107	STR5
8	STR6	28	NZ	48	PT	68	NZ	88	DELE	108	NZ
9	DELR	29	NY/S1	49	DD1	69	NY/S1	89	DD2	109	NY/S1
10	STR4	30	STR2	50	PITCH	70	STR4	90	STR2	110	PITCH
11	STR5	31	STR3	51	YAW	71	STR5	91	STR3	111	YAW
12	NZ	32	STR6	52	NZ	72	DELF	92	NZ	112	DELE
13	NY/S1	33	DELR	53	NY/S1	73	DELNG	93	NY/S1	113	E5-9
14	PITCH	34	STR4	54	STR2	74	PITCH	94	STR4	114	STR2
15	YAW	35	STR5	55	STR3	75	YAW	95	STR5	115	STR3
16	DELE	36	NZ	56	STR6	76	NZ	96	DD3	116	NZ
17	VG	37	SY/S1	57	DELR	77	NY/S1	97	DD4	117	NY/S1
18	STR2	38	PITCH	58	STR4	78	STR2	98	PITCH	118	STR4
19	STR3	39	YAW	59	STR5	79	STR3	99	YAW	119	STR5
20	NZ	40	DELE	60	NZ	80	STR6	100	NZ	120	DELF

TABLE 2.3 (Continued)

ECU-67/A MULTIPLEXING FORMAT C-141A/C-130

Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter
121	DELNG	141	NY/S1	161	VG	181	NY/S1	201	DELR	221	NY/S1
122	PITCH	142	STR4	162	STR2	182	PITCH	202	STR4	222	STR2
123	YAW	143	STR5	163	STR3	183	YAW	203	STR5	223	STR3
124	NZ	144	DD6	164	NZ	184	DELE	204	NZ	224	STR6
125	NY/S1	145	DD7	165	NY/S1	185	DD3	205	NY/S1	225	DELR
126	STR2	146	PITCH	166	STR4	186	STR2	206	PITCH	226	STR4
127	STR3	147	YAW	167	STR5	187	STR3	207	YAW	227	STR5
128	STR6	148	NZ	168	DELF	188	NZ	208	DELE	228	NZ
129	DELR	149	NY/S1	169	DELNG	189	NY/S1	209	DELP	229	NY/S1
130	STR4	150	STR2	170	PITCH	190	STR4	210	STR2	230	PITCH
131	STR5	151	STR3	171	YAW	191	STR5	211	STR3	231	YAW
132	NZ	152	STR6	172	NZ	192	DD9	212	NZ	232	DELE
133	NY/S1	153	DELR	173	NY/S1	193	DD10	213	NY/S1	233	DD11
134	PITCH	154	STR4	174	STR2	194	PITCH	214	STR4	234	STR2
135	YAW	155	STR5	175	STR3	195	YAW	215	STR5	235	STR3
136	DELE	156	NZ	176	STR6	196	NZ	216	DELF	236	NZ
137	DD5	157	NY/S1	177	DELR	197	NY/S1	217	DELNG	237	NY/S1
138	STR2	158	PITCH	178	STR4	198	STR2	218	PITCH	238	STR4
139	STR3	159	YAW	179	STR5	199	STR3	219	YAW	239	STR5
140	NZ	160	DELE	180	NZ	200	STR6	220	NZ	240	DD12

TABLE 2.3 (Continued)

ECU-67/A MULTIPLEXING FORMAT C-141A/C-130

Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter	Word No.	Para-meter
241	CTR	261	NY/S1	281	PS	301	NY/S1	321	DELR	341	NY/S1
242	PITCH	262	STR4	282	STR2	302	PITCH	322	STR4	342	STR2
243	YAW	263	STR5	283	STR3	303	YAW	323	STR5	343	STR3
244	NZ	264	DELF	284	NZ	304	DELE	324	NZ	344	STR6
245	NY/S1	265	DELNG	285	NY/S1	305	E1-4	325	NY/S1	345	DELR
246	STR2	266	PITCH	286	STR4	306	STR2	326	PITCH	346	STR4
247	STR3	267	YAW	287	STR5	307	STR3	327	YAW	347	STR5
248	STR6	268	NZ	288	PT	308	NZ	328	DELE	348	NZ
249	DELR	269	NY/S1	289	DD1	309	NY/S1	329	DD2	349	NY/S1
250	STR4	270	STR2	290	PITCH	310	STR4	330	STR2	350	PITCH
251	STR5	271	STR3	291	YAW	311	STR5	331	STR3	351	YAW
252	NZ	272	STR6	292	NZ	312	DELF	332	NZ	352	DELE
253	NY/S1	273	DELR	293	NY/S1	313	DELNG	333	NY/S1	353	E5-9
254	PITCH	274	STR4	294	STR2	314	PITCH	334	STR4	354	STR2
255	YAW	275	STR5	295	STR3	315	YAW	335	STR5	355	STR3
256	DELE	276	NZ	296	STR6	316	NZ	336	DD3	356	NZ
257	VG	277	NY/S1	297	DELR	317	NY/S1	337	DD4	357	NY/S1
258	STR2	278	PITCH	298	STR4	318	STR2	338	PITCH	358	STR4
259	STR3	279	YAW	299	STR5	319	STR3	339	YAW	359	STR5
260	NZ	280	DELE	300	NZ	320	STR6	340	NZ	360	DELF

TABLE 2.3 (Concluded)

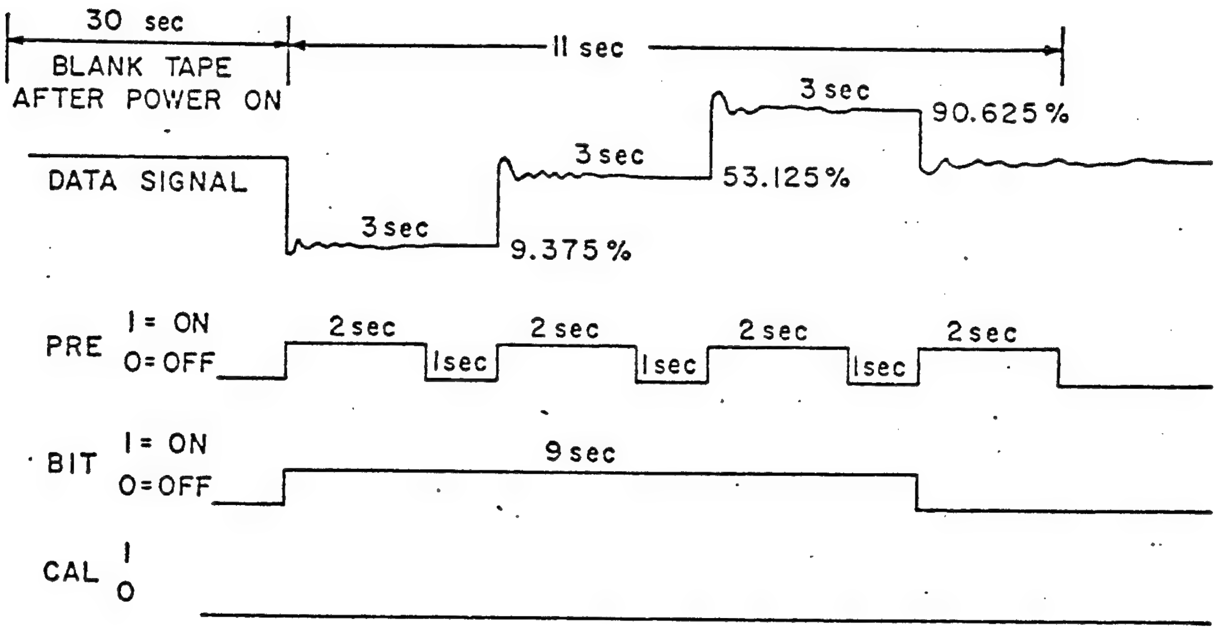
ECU-67/A MULTIPLEXING FORMAT C-141A/C-130

Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter	Word No.	Parameter
361	DELNG	381	NY/S1	401	VG	421	NY/S1	441	DELR	461	NY/S1
362	PITCH	382	STR4	402	STR2	422	PITCH	442	STR4	462	STR2
363	YAW	383	STR5	403	STR3	423	YAW	443	STR5	463	STR3
364	NZ	384	DD6	404	NZ	424	DELE	444	NZ	464	STR6
365	NY/S1	385	DD7	405	NY/S1	425	DD8	445	NY/S1	465	DELR
366	STR2	386	PITCH	406	STR4	426	STR2	446	PITCH	466	STR4
367	STR3	387	YAW	407	STR5	427	STR3	447	YAW	467	STR5
368	STR6	388	NZ	408	DELF	428	NZ	448	DELE	468	NZ
369	DELR	389	NY/S1	409	DELNG	429	NY/S1	449	DELP	469	NY/S1
370	STR4	390	STR2	410	PITCH	430	STR4	450	STR2	470	PITCH
371	STR5	391	STR3	411	YAW	431	STR5	451	STR3	471	YAW
372	NZ	392	STR6	412	NZ	432	DD9	452	NZ	472	DELE
373	NY/S1	393	DELR	413	NY/S1	433	DD10	453	NY/S1	473	DD11
374	PITCH	394	STR4	414	STR2	434	PITCH	454	STR4	474	STR2
375	YAW	395	STR3	415	STR3	435	YAW	455	STR5	475	STR3
376	DELE	396	NZ	416	STR6	436	NZ	456	DELF	476	NZ
377	DD5	397	NY/S1	417	DELR	437	NY/S1	457	DELNG	477	NY/S1
378	STR2	398	PITCH	418	STR4	438	STR2	458	PITCH	478	STR4
379	STR3	399	YAW	419	STR5	439	STR3	459	YAW	479	STR5
380	NZ	400	DELE	420	NZ	440	STR6	460	NZ	480	DD12

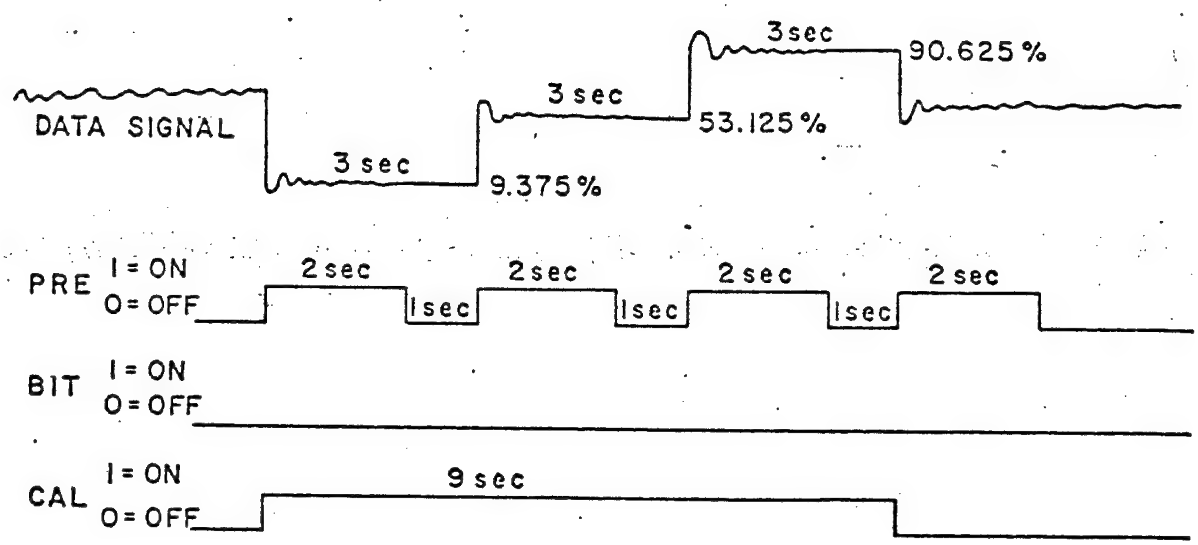
carefully noted by the reader! Sample 1 as indicated in Table 2.2 and Table 2.3 is really the one-byte tape gap on the cartridge tape and is not written on the R/T output tape, thus, even though Table 2.2 and Table 2.3 indicate 480 data samples, only 479 (plus the additional 7) are actually written out.

A description of the notations used in Table 2.2 is as follows:

T/CTR	=	frame counter (0-255)
V_i	=	differential pressure
PLA	=	power lever angle
δ_{SB}	=	speed brake position
FCTR	=	fuel counter
N_Y	=	lateral acceleration
σ	=	strain
r	=	yaw rate
p	=	roll rate
q	=	pitch rate
N_z	=	normal acceleration
$E_1 - E_4$	=	events 1-4 stored in 4 high order bits
$E_5 - E_9$	=	events 5-9 stored in 5 high order bits
DD1-DD12	=	documentary data (see Table 2.4)
h_p	=	static pressure
STR1-STR6	=	strain 1 - strain 6
DELE	=	elevator deflection
VG	=	ground speed
DELF	=	flap deflection
DELNG	=	nose gear angle
PS	=	static pressure
PT	=	total pressure
DELR	=	rudder deflection
DELP	=	cabin pressure



(a) BUILT-IN TEST (BIT) CYCLE



(b) CALIBRATION TEST (CAL) CYCLE

Figure 2.1. BIT and CAL Cycles.

TABLE 2.4

DOCUMENTARY DATA ENCODER

Discrete Data	Number of Decimal Digits	Programmer Samples for BCD Formatting (Ref. MIL-C-83166 Slash Specifications)	
		1st Sample	2nd Sample
Aircraft serial number	4	DD_1	DD_2
Gross weight	4	DD_3	DD_4
Fuel weight	4	DD_5	DD_6
Base	2	DD_7	
Mission	2	DD_8	
Configuration	3	DD_9 DD_{10}(2nd half)	*DD_{10}(1st half)
Date	5	DD_{11}**	DD_{12}***

*The first digit of DD_{10} is the third digit of configuration and the second digit of DD_{10} is the last digit of the year

DD_{10} 1st half				DD_{10} 2nd half			
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
third digit of configuration				last digit of year			

**This sample also contains the BIT, CAL, and PRE indicators

DD_{11}

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
BIT	CAL	PRE	Tens digits of month	Unit digits of month			

***This sample is broken down as follows:

DD_{12}

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Power on	AGE on	Tens digit of day		Units digit of day			

NOTE: The bits in the above samples are represented as powers of two for clarity.

The CAL cycle, as shown in Figure 2.1, is very similar to the BIT cycle. No blank tape is run off, however. This calibration cycle is triggered every 90 gaps (or 3 minutes) by an automatic system unless an auto-cal disable switch on the recorder control panel is turned to "off". This automatic cal is normally not used due to an excessive loss of data.

For each indicator, BIT, CAL, and PRE, a value of one indicates it is on and a value of 0 indicates it is off.

In summary, the computer input data tape contains frames having 486 8-bit bytes. Five frames of 486 8-bit bytes make up one physical record. The reformatted tape should begin with a frame of header information followed by a BIT cycle. The end-of-flight is designated by all ones in the first four bits of byte 480 of the frame. The last frame of data on the tape is followed by at least one complete frame with all bits on. Following the end of the last physical record would be an octal 17 end of file mark.

2.2 CARD INPUT

Card input for the four compression programs varies because of the number of parameters being examined and because of the two different methods of data compression. Therefore, the inputs are specified for each of the programs in the following paragraphs. The programs are COMPA1, COMPA2, COMPB1, and COMPB2 representing method A for the A-10, Method A for the C-130/C-141, Method B for the A-10, and Method B for the C-130/C-141 respectively.

2.2.1 Input to COMPA1

This is used for Method A on the A10 tape format:

Column	Description	Format
1-4	Aircraft Title	A4
5	Process Ground Data. Enter: 0 or blank - do not wish to examine ground data 1 - examine all data on tape	I1
6-29	Threshold Values. The eight thresholds are in the following order: NZ, NY, P, Q, R, ALT, A/S, FUEL	8I3

30-44	Constant Mean Values. The five mean(normal) values are in the following order: NZ, NY, P, Q, R	5I3
45-80	Blank	

2.2.2 Input to COMPA2

This is used for the Method A on the C130 & C141 tape format:

Column	Description	Format
1-4	Aircraft Title	A4
5	Process Ground Data. Enter: 0 or blank - do not wish to examine ground data 1 - examine all data on tape	I1
6-23	Threshold values. The six thresholds are in the following order: NZ, NY/S1*, Q, R, ALT, and AS.	6I3
24-35	Constant Mean Values. Digit numbers are as follows: NZ, NY/S1, Q, R	4I3
36-80	Blank	

*NOTE: S1 is recorded in place of NY on the C-130. A value of 999 must be used for the S1 threshold to exclude it from the compression process.

2.2.3 Input to COMPB1

This is used for Method B on the A10 tape format.

Column	Description	Format
1-4	Aircraft Title	A4
5	Process Ground Data. Enter: 0 or blank - do not wish to examine ground data 1 - wish to examine all data recorded	I1
6-50	Window Value. The fifteen values are in the following order: NY, NZ, P, Q, R, $\dot{P}$, $\dot{Q}$, $\dot{R}$, FCTR, PLA, δ_{SB}, σ, hp, and V_i	14I3
51-80	Blank	

2.2.4 Input to COMPB2

This is used for Method B on the C130 and C141 tape format:

Column	Description	Format
1-4	Aircraft Title	A4
5	Process Ground Data. Enter: 0 or blank - do not wish to examine ground data 1 - examine all data recorded on tape	I1
6-62	Window Values. The nineteen values are in the following order: NY/S1*, NZ, Q, R, $\dot{Q}$, $\dot{R}$, S2, S3, S4, S5, S6, DELR, DELE, DELF, DELNG, VG, PS, PT, DELP *C130 - NY is replaced by the S1 parameter	19I3
63-80	Blank	

SECTION 3
DESCRIPTION OF PROGRAMS

Four separate programs were written for the evaluation of the two methods of data compression. A separate program was written for use on the A-10 data and the C-130/C-141 data for each of the compression techniques. This was done to keep the programming simpler in order to make a more accurate comparison of run times. Each of these four programs will be described in the following paragraphs followed by a description of the general subroutines PACK and UNPACK which are used in all of the programs.

Each of the four programs must do a certain amount of editing of the raw data. After program initialization and the program control cards are read, the data is processed by reading five frames or ten seconds of data from the RT tape. The very first frame from each tape, the header frame, is ignored. Thereafter, each frame is handled as an entity. The frame is first examined for parity errors as indicated by a non-zero value in bytes 483 and 484 combined. If a parity error is detected, the frame is counted and then ignored for further processing. The next check is made for a short record which is indicated by the four least significant bits of byte 480 containing something other than a 9. If a short record is detected, it is counted and ignored. A check is then made for the presence of a BIT, CAL, or PRE indicator in either half of the frame. This is indicated by the presence of a one in the appropriate bits of DD11 as indicated in Table 2.4. If the BIT, CAL, or PRE bit is on, the second of data is counted and then ignored with the exception of the BIT bit being on. This condition is necessary for the beginning of a flight if the aircraft is not airborne. If the aircraft is airborne, the data is ignored. If the BIT bit is on exclusively, a single sample of each of the parameters is saved for inclusion in the header record as calibration data. Each of the three calibration levels is handled in this manner. The first occurrence of a BIT cycle also signals the end of a previous flight and an end of flight record is written. A BIT cycle can be initiated by a power interuption while the aircraft is airborne which will be ignored by the program.

If the data is valid data, having passed all of the previous tests, a check is made to determine whether the aircraft is airborne. If the aircraft is airborne, the specific compression technique is applied. If the aircraft is not in the air and the decision has been made to not process ground data, the second of data is ignored; otherwise the data is processed.

The initial values of airspeed, altitude, fuel counter and events are obtained from the first valid data record following a BIT cycle for Method A and for all parameters for Method B.

If no exceedance or window crossings are found in a segment of tape between BIT cycles, there will not be a header record or any other evidence of the segment on the output tape. If, however, ground data is being processed and exceedances or window crossings are found, the segment will be designated as a flight even though there is no inflight data in the segment.

If the data is good data, having passed all of the previous tests, a check is made to determine whether the aircraft is airborne. If the aircraft is airborne, the specific compression technique is applied. If the aircraft is not in the air and the decision has been made to not process ground data, the second of data is ignored; otherwise the data is processed.

The initial values of airspeed, altitude, fuel counter and events are obtained from the first good data record following a BIT cycle for Method A and for all parameters for Method B.

3.1 METHOD A

Data compression using Method A is accomplished by testing selected parameters for exceedance of predetermined thresholds. Specifically, if $v_m - T_v > v_i$ or $v_i > v_m + T_v$

where v_m = normal value

T_v = threshold,

then the data is considered active and the entire second of data is written to the output tape. If the data is not determined active by this criterion, a further check is made for a change in

flight condition. A flight condition change is defined as a change by more than a preselected value of altitude, airspeed, etc. from the last change in that parameter. A change in any event is also considered a flight condition change. If a change in flight condition occurs, the entire second of data is written to the output tape. One second of additional data is written to the output tape following a period of active data as determined by either of the prior criteria.

For each second of data written to the output tape, computations of the roll, pitch and yaw acceleration are made, where applicable, using the following equation

$$\dot{P} = \frac{S}{12} [(P_{i-2} - P_{i+2}) + 8(P_{i+1} - P_{i-1})]$$

where $\dot{P}$ = roll, pitch or yaw acceleration (counts/sec^2)

S = sample rate (samples/sec)

P = roll, pitch or yaw rate (counts)

The accelerations are not computed for the first two or the last two samples in each continuous interval of active data. The acceleration values are biased by 128 to avoid a negative sign which would require an additional bit for storage.

A small peculiarity of the above equation should be noted. Each time there is a change in the acceleration from the zero position, the first value computed is of the wrong sign. For example, if we have four consecutive values of 128 followed by a 129, the acceleration computed from the equation is -1.25 counts/sec for a sampling rate of 15 samples/sec. This is due to the fact that the quartic equation must pass through all five points of the curve. This should, however, present no real problem since the magnitude of these values should be quite small and well below the threshold of interest.

All recorded parameters need not be examined for activity with this compression technique since all data is retained if any of the analyzed parameters become active.

3.1.1 COMPA1 Program

The COMPA1 program was written to compress data recorded on the A-10 aircraft specifically. The parameters examined for threshold exceedances are N_y, N_z, pitch, roll and yaw, all of which are sampled at a rate of 15 samples/sec. Flight condition parameters are altitude, airspeed, fuel counter and the two event samples all of which are sampled at the rate of one sample/sec with the exception of the fuel counter which is sampled at 15 samples/sec. Inflight data on the A-10 aircraft is determined by the presence of a one in event 3.

3.1.2 COMPA2 Program

The COMPA2 program will work on both C-130 and C-141 data. However, when running C-130 data the threshold value for S1, which is recorded in place of N_y, must be set to a large number to avoid processing the strain channel. A strain channel, unless it is at some location which has a constant mean stress, cannot be processed by looking for excursions from a constant mean.

The parameters that are examined for threshold exceedances are N_y, N_z, pitch and yaw. N_y and N_z are sampled at 30 times/sec while pitch and yaw are sampled at 20 times/sec. Flight condition parameters are altitude, airspeed and the two event samples which are sampled at 1 sample/sec. If the aircraft is not airborne and ground data are being processed, ground speed becomes the only flight condition parameter. Ground speed is sampled at two samples/sec. Inflight data on both the C-130 and C-141 aircraft is determined by the presence of a one in the first event.

3.2 METHOD B

The technique of data compression designated as Method B involves the location of window crossings and peaks on selected parameters. A window crossing is determined by the following equation

$$v_i - w > v \quad \text{or} \quad v > v_i + w$$

where v = present value

v_i = last stored value

w = window value

Upon detecting a window crossing the value along with a parameter ID and time of occurrence are written to the output tape. If a minimum or a maximum is reached by the parameter between window crossings, it is also written to the output tape. A maximum is generated by a change from positive slope of two or more window crossings to a negative slope and a minimum is generated by a change from negative to positive slope.

All parameters of interest are handled in the same manner with this compression technique, however, window values cannot be varied for the events which are treated as combined values of events 1-4 and events 1-9. The event windows are set, within the program, to zero.

A continuous computation of the roll, pitch, and yaw accelerations, where applicable, is made using all samples of roll, pitch and yaw. These accelerations are then analyzed for window crossings and peaks in the same way as the other parameters. An input option allows for processing or ignoring ground data at the discretion of the user.

A major difference between the two methods of data compression is that in using Method B all parameters of interest must be included in the analysis for window crossings since coincident values of other recorded parameters are not saved when activity occurs on one parameter. Therefore, if a parameter is not included in the analysis, that data is lost during the compression process.

3.2.1 Program COMPB1

The COMPB1 program was written to process the A-10 data using the Method B technique. The parameters that are examined for window crossings are: N_y, N_z, pitch, roll, yaw, fuel counter, speed brake position, power lever angle, strain, altitude,

TABLE 3.1

SAMPLING RATES OF PARAMETERS FOR C-130/C-141

Parameter	Sampling Rate (Samples/sec) C-130	C-141
Pitch	20	20
Yaw	20	20
N_z	30	30
N_y		30
S1	30	
S2	20	20
S3	20	20
S4	20	20
S5	20	20
S6	10	10
DELR	10	10
DELE	10	10
DELF	5	5
DELNG	5	5
PS	1	1
PT	1	1
VG	2	2
DELP	1	1
E1-9	1	1

airspeed, and events 1-9. The pitch, roll, and yaw accelerations are computed and analyzed. The altitude, airspeed and events are sampled at 1 sample/sec and all of the other parameters at 15 samples/sec.

3.2.2 Program COMPB2

The COMPB2 program was written to process C-130/C-141 data. The C-141 parameters analyzed for window crossings and peaks are N_y, N_z, pitch, yaw, strains 2-6, rudder deflection, elevator deflection, flap deflection, nose gear angle, ground speed, cabin pressure, altitude, airspeed, and events 1-9. The same parameters, with the exception of strain 1 replacing N_y, are analyzed for the C-130 data. The sampling rates for the various parameters are shown in Table 3.1. Airborne data is detected by the presence of a one in event 1 for both aircraft. In addition to the recorded parameters, the pitch and yaw accelerations are computed continuously and are monitored for window crossings and peaks.

3.3 SUBROUTINE PACK

Subroutine PACK is a generalized routine for packing data into a condensed format for output to tape. The routine is capable of extracting any number of bits from any number of unpacked words and consecutively packing these into an array of 60 bit packed words. The parameters in the call statement to subroutine PACK and their usage are as follows:

- UP - array containing the unpacked data
- NUP - number of words from UP to be packed
- P - array to receive the packed data
- J - first available word in the P array to receive packed data
- NB - first available bit in the Jth word of P array to receive packed data ($1 \leq NB \leq 60$)
- ITBL - array of length NUP containing the number of bits from each word in UP to be packed into P (if positive bits are to be taken from right side of packed word and if negative from the left side of packed word).

The values of J and NB are saved internally within PACK so that in a subsequent call to the routine it will continue packing into the P array where it left off, if the values of J and NB have not been altered in the calling program.

3.4 SUBROUTINE UNPACK

Subroutine UNPACK is a generalized routine for unpacking data into 60 bit words. In the present application it is used to unpack the 486 consecutive 8 bit samples from a frame of RT data. The parameters in the call to PACK and their usage are as follows:

- UP - array to contain the unpacked data.
- NUP - number of words in UP the unpacked data will occupy
- P - array containing the packed data
- J - word in the P array at which unpacking is to begin
- NB - bit in the Jth word of the P array at which unpacking is to begin
- ITBL - array of length NUP containing the number of bits to unpack into each word of the UP array (if positive, bits are to be right justified with leading bits zero filled and if negative, left justified with trailing bits zero filled unless ITBL is a multiple of 6 in which case trailing bits are blank filled)

The variation in the possible values contained in the ITBL array allows for unpacking integer, real or alphanumeric information.

The values of J and NB are saved internally within UNPACK so that in a subsequent call to the routine it will continue to unpack from the P array at the point it left off, if the values of J and NB have not been altered by the calling program.

SECTION 4
PROGRAM OUTPUT

The output from the data compression programs consists of printed output and tape. The printed output lists the input parameters, a summary of each flight encountered on the RT tape and a summary of the total tape. Each time a BIT cycle is encountered, all counters are reinitialized and the summary data for a flight is printed if appropriate. A flight on the output tape is either an actual flight or a segment of ground data depending upon whether ground data is being processed. If ground data is being processed, data will be written to the output tape only if at least one exceedance or window crossing was detected. If ground data is not being processed, only the data occurring inflight will be written to the output tape and if no lift-off occurs during the segment, no trace of the segment will occur on the output tape or the printed output.

The printed output from programs COMPA1 and COMPA2 using Method A of data compression are identical with the exception of the parameters analyzed and their thresholds. A detailed explanation, using output from COMPA1, is shown in Figure 4.1. The printed output from programs COMPB1 and COMPB2 are also identical with the exception of the parameters analyzed and their window values. The output from COMPB1 is used in Figure 4.2 to provide a detailed explanation.

The tape output of the four compression programs is different for each program. All of the programs write flight header records (Type 1) followed by data records (Type 2) and end of flight records (Type 3). The difference in the formats of these records is due to the compression technique and the number of parameters being analyzed. The detailed format of the three record types for each of the compression programs is shown in Tables 4.1, 4.2, 4.3, and 4.4. Although it might be desirable from the standpoint of generalization to standardize the output formats for a given compression technique, considerable savings in output space is

realized in the case of the A-10 using Method A with the present format. This is due to the large number of vacant samples. In fact, a 37 percent compression of the data can be realized by only removing the unused samples from the data.

```
AIRCRAFT:   A10
PROCESSING GROUND DATA
THRESHOLDS: NZ= NY=  P=  Q=  R=ALT= AS= FC
            11  13   5   6   6   9   4   4
VALUES OF: NZ= NY=  P=  Q=  R
           85 128 128 128 128

FLIGHT#  1 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS PRINTED:   2501
2.  SECONDS OF GROUND DATA:    402
3.  SECONDS OF ACTIVE DATA:   1526
4.  TOTAL NUMBER OF SECONDS:   4456

FLIGHT#  2 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS PRINTED:   6213
2.  SECONDS OF GROUND DATA:   3085
3.  SECONDS OF ACTIVE DATA:   4448
4.  TOTAL NUMBER OF SECONDS:  10376

FLIGHT#  3 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS PRINTED:   3940
2.  SECONDS OF GROUND DATA:   2111
3.  SECONDS OF ACTIVE DATA:   2335
4.  TOTAL NUMBER OF SECONDS:   6318

FLIGHT#  4 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS PRINTED:   5635
2.  SECONDS OF GROUND DATA:   1010
3.  SECONDS OF ACTIVE DATA:   4164
4.  TOTAL NUMBER OF SECONDS:   8170

FLIGHT#  5 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS PRINTED:    921
2.  SECONDS OF GROUND DATA:   1910
3.  SECONDS OF ACTIVE DATA:    545
4.  TOTAL NUMBER OF SECONDS:   1974

FLIGHT#  6 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS PRINTED:   4111
2.  SECONDS OF GROUND DATA:   1504
3.  SECONDS OF ACTIVE DATA:   2911
4.  TOTAL NUMBER OF SECONDS:   8152

FLIGHT#  7 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS PRINTED:   5043
2.  SECONDS OF GROUND DATA:   1922
3.  SECONDS OF ACTIVE DATA:   3690
4.  TOTAL NUMBER OF SECONDS:   9090

FLIGHT#  8 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS PRINTED:    277
2.  SECONDS OF GROUND DATA:    660
3.  SECONDS OF ACTIVE DATA:    158
4.  TOTAL NUMBER OF SECONDS:    682
```

Figure 4.1. Method A Output.

```
PRINTOUT FOR COMPA1 STATISTICS.

1. THE NUMBER OF TYPE1 RECORDS PRINTED:        8

2. THE NUMBER OF TYPE2 RECORDS PRINTED:    28641

3. THE NUMBER OF TYPE3 RECORDS PRINTED:        8

4. SECONDS OF GROUND DATA:  16018

5. TOTAL ACTIVE SECONDS:  19777

6. TOTAL NUMBER OF SECONDS:  52810
EOJ
```

Figure 4.1 (Continued)

Notes:

1. The Number of Type 2 Records Printed

The number of seconds of data found where a parameter exceeded the input threshold plus the additional second written at the end of each active period.

2. Seconds of Ground Data

The number of recorded seconds while the aircraft was not airborne.

3. Seconds of Active Data

The number of seconds of data found where a parameter exceeded threshold.

4. Total Number of Seconds

The total time for the flight from recorder start to stop.

5. The Number of Type 1 Records Printed

The total number of flight header records written to the output tape. This is the number of flights encountered on the tape.

6. The Number of Type 3 Records Printed

The total number of end of flight records written to the output tape.

Figure 4.1. (Concluded)

```
AIRCRAFT:    A10
PROCESSING GROUND DATA
WINDOW VALUES: NY   NZ   P    Q    R   P-DOT Q-DOT R-DOT
               13   11   5    6    6    20     20    20
              FCTR FLA CSB SIGMA ALT  AS
                4   1   1    5    9   4

FLIGHT#  1 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS WRITTEN:     106
2.  SECONDS OF GROUND DATA:      402
3.  SAMPLES OF ACTIVE DATA:     2419
4.  TOTAL NUMBER OF SECONDS:    [illegible]456

FLIGHT#  2 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS WRITTEN:     906
2.  SECONDS OF GROUND DATA:     3085
3.  SAMPLES OF ACTIVE DATA:    20838
4.  TOTAL NUMBER OF SECONDS:   10376

FLIGHT#  3 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS WRITTEN:     130
2.  SECONDS OF GROUND DATA:     2111
3.  SAMPLES OF ACTIVE DATA:     2982
4.  TOTAL NUMBER OF SECONDS:    6318

FLIGHT#  4 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS WRITTEN:    1581
2.  SECONDS OF GROUND DATA:     1010
3.  SAMPLES OF ACTIVE DATA:    36345
4.  TOTAL NUMBER OF SECONDS:    8170

FLIGHT#  5 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS WRITTEN:      55
2.  SECONDS OF GROUND DATA:     2554
3.  SAMPLES OF ACTIVE DATA:     1260
4.  TOTAL NUMBER OF SECONDS:    2646

FLIGHT#  6 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS WRITTEN:     366
2.  SECONDS OF GROUND DATA:     1504
3.  SAMPLES OF ACTIVE DATA:     8409
4.  TOTAL NUMBER OF SECONDS:    8152

FLIGHT#  7 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS WRITTEN:     811
2.  SECONDS OF GROUND DATA:     1922
3.  SAMPLES OF ACTIVE DATA:    18645
4.  TOTAL NUMBER OF SECONDS:    9630

FLIGHT#  8 STATISTICS
1.  THE NUMBER OF TYPE2 RECORDS WRITTEN:      14
2.  SECONDS OF GROUND DATA:     1524
3.  SAMPLES OF ACTIVE DATA:      307
4.  TOTAL NUMBER OF SECONDS:    1532
```

Figure 4.2. Method B Output.

```
PRINTOUT FOR COMPB1 STATISTICS:

1.  THE NUMBER OF TYPE1 RECORDS WRITTEN:        8

2.  THE NUMBER OF TYPE2 RECORDS WRITTEN:     3969

3.  THE NUMBER OF TYPE3 RECORDS WRITTEN:        0

4.  SECONDS OF GROUND DATA:  16018

5.  TOTAL ACTIVE SAMPLES:  91205

6.  THE NUMBER OF PEAKS STORED:     31091

7.  THE TOTAL NUMBER OF SECONDS:      52810
```

Figure 4.2 (Continued)

Notes:

1. The Number of Type 2 Records Written

This is the number of active data records written. Each record, with the exception of the last for each flight, contains 23 window crossings and peaks.

2. Seconds of Ground Data

The number of recorded seconds where the aircraft was not airborne.

3. Samples of Active Data

The number of window crossings and peaks found in the flight.

4. Total Number of Seconds

The total time for the flight from recorder start to stop.

5. The Number of Type 1 Records Written

The total number of flight header records written to the output tape. This is the number of flights encountered on the tape.

6. The Number of Type 3 Records Written

The total number of end of flight records written to the output tape.

7. The Number of Peaks Stored

This is the total number of actual peaks encountered on the tape.

Figure 4.2 (Concluded)

TABLE 4.1

COMPA1 TAPE OUTPUT FORMAT

RECORD TYPE 1

Data Item	Bits	Word
Record type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
ALT threshold	8	4
AS threshold	8	
Fuel threshold	8	
NY threshold	8	
NZ threshold	8	
P threshold	8	
Q threshold	8	
Blank	4	
R threshold	8	5
NY Cal 1	8	
NZ Cal 1	8	
P Cal 1	8	
Q Cal 1	8	
R Cal 1	8	

TABLE 4.1 (Continued)

COMPA1 TAPE OUTPUT FORMAT

RECORD TYPE 1 (Continued)

Data Item	Bits	Word
Delta-SB Cal 1	8	
Blank	4	
Sigma Cal 1	8	6
PLA Cal 1	8	
Airspeed Cal 1	8	
Altitude Cal 1	8	
Fuel Counter Cal 1	8	
NY Cal 2	8	
NZ Cal 2	8	
Blank	4	
P Cal 2	8	7
Q Cal 2	8	
R Cal 2	8	
Delta SB Cal 2	8	
Sigma Cal 2	8	
PLA Cal 2	8	
Airspeed Cal 2	8	
Blank	4	
Altitude Cal 2	8	8
Fuel Counter Cal 2	8	
NY Cal 3	8	
NZ Cal 3	8	
P Cal 3	8	
Q Cal 3	8	
R Cal 3	8	
Blank	4	
Delta SB Cal 3	8	9
Sigma Cal 3	8	
PLA Cal 3	8	
Airspeed Cal 3	8	

TABLE 4.1 (Continued)

COMPA1 TAPE OUTPUT FORMAT

RECORD TYPE 1 (Continued)

Data Item	Bits	Word
Altitude Cal 3	8	
Fuel Counter Cal 3	8	
Blank	12	
Blank	960	10-25

TABLE 4.1 (Continued)

COMPAL TAPE OUTPUT FORMAT

RECORD TYPE 2

Data Item	Bits	Word	
Record type	4	1	
Time*	16		
Airspeed	8		
Altitude	8		
Fuel Counter	8		
E1-4	8		
E5-9	8		
NY	8		Repeat 15 times
NZ	8		
P	8		
Q	8		
R	8		
$\dot{P}$	8		
$\dot{Q}$	8		
$\dot{R}$	8		
Delta SB	8		
Sigma	8		
PLA	8		
Blank	8		
	8 x 180 = 1440		
	TOTAL	25 words	

*Time is from start of recorder.

TABLE 4.1 (Concluded)

COMPA1 TAPE OUTPUT FORMAT

RECORD TYPE 3

Data Item	Bits	Word
Record type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DDL	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD0	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Flight duration (in seconds)	60	4
Blank	1260	5-25

TABLE 4.2

COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 1

Data Item	Bits	Word
Record Type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
ALT threshold	8	4
AS threshold	8	
NY threshold	8	
NZ threshold	8	
Q threshold	8	
R Threshold	8	
Blank	12	
Q Cal 1	8	5
R Cal 1	8	
NZ Cal 1	8	
NY - (S1) Cal 1	8	
S2 Cal 1	8	
S3 Cal 1	8	
S6 Cal 1	8	
Blank	4	

TABLE 4.2 (Continued)

COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 1 (Continued)

Data Item	Bits	Word
DELR Cal 1	8	6
S4 Cal 1	8	
S5 Cal 1	8	
DELE Cal 1	8	
VG Cal 1	8	
DELF Cal 1	8	
DELNG Cal 1	8	
Blank	4	
PS Cal 1	8	7
PT Cal 1	8	
DELP Cal 1	8	
Q Cal 2	8	
R Cal 2	8	
NZ Cal 2	8	
NY Cal 2	8	
Blank	4	
S2 Cal 2	8	8
S3 Cal 2	8	
S6 Cal 2	8	
DELR Cal 2	8	
S4 Cal 2	8	
S5 Cal 2	8	
DELE Cal 2	8	
Blank	4	
VG Cal 2	8	9
DELF Cal 2	8	
DELNG Cal 2	8	
PS Cal 2	8	
PT Cal 2	8	
DELP Cal 2	8	
Q Cal 3	8	
Blank	4	

TABLE 4.2 (Continued)

COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 1 (Continued)

Data Item	Bits	Word
R Cal 3	8	10
NZ Cal 3	8	
NY Cal 3	8	
S2 Cal 3	8	
S3 Cal 3	8	
S6 Cal 3	8	
DELR Cal 3	8	
Blank	4	
S4 Cal 3	8	11
S5 Cal 3	8	
DELE Cal 3	8	
VG Cal 3	8	
DELF Cal 3	8	
DELNG Cal 3	8	
PS Cal 3	8	
Blank	4	
PT Cal 3	8	12
DELP Cal 3	8	
Blank	44	
Blank	1560	13-38

TABLE 4.2 (Continued)

COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 2

Data Item	Bits	Word
Record type	4	1
Time *	16	
One second of data as is	8 x 239 = 1912	
$\dot{Q}$	8 x 20 = 160	
$\dot{R}$	8 x 20 = 160	
Blank	28	
	TOTAL	38 words

*Time is from start of recorder

TABLE 4.2 (Concluded)

COMPA2 TAPE OUTPUT FORMAT

RECORD TYPE 3

Data Item	Bits	Word
Record Type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Flight Duration (time in seconds)	60	4
Blank	2040	5-38

TABLE 4.3

COMPB1 TAPE OUTPUT FORMAT
RECORD TYPE 1

Data Item	Bits	Word
Record Type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
NY Window	8	4
NZ Window	8	
P Window	8	
Q Window	8	
R Window	8	
P-DOT Window	8	
Q-DOT Window	8	
Blank	4	
R-DOT Window	8	5
Delta SB Window	8	
Sigma Window	8	
Delta PLA Window	8	
Airspeed Window	8	
Altitude Window	8	
Fuel Counter Window	8	
Blank	4	
NY Cal 1	8	6
NZ Cal 1	8	

TABLE 4.3 (Continued)

COMPB1 TAPE OUTPUT FORMAT
RECORD TYPE 1

Data Item	Bits	Word
P Cal 1	8	
Q Cal 1	8	
R Cal 1	8	
Delta SB Cal 1	8	
Sigma Cal 1	8	
Blank	4	
Delta PLA Cal 1	8	7
Airspeed Cal 1	8	
Altitude Cal 1	8	
Fuel Counter Cal 1	8	
NY Cal2	8	
NZ Cal2	8	
P Cal2	8	
Blank	4	
Q Cal2	8	8
R Cal2	8	
Delta SB Cal2	8	
Sigma Cal2	8	
Delta PLA Cal2	8	
Airspeed Cal2	8	
Altitude Cal2	8	
Blank	4	
Fuel Counter Cal2	8	9
NY Cal3	8	
NZ Cal3	8	
P Cal3	8	
Q Cal3	8	
R Cal3	8	
Delta SB Cal3	8	
Blank	4	

TABLE 4.3 (Continued)

COMPB1 TAPE OUTPUT FORMAT
RECORD TYPE 1

Data Item	Bits	Word
Sigma Cal3	8	10
Delta PLA Cal3	8	
Airspeed Cal3	8	
Altitude Cal3	8	
Fuel Counter Cal3	8	
Blank	20	
Blank	360	11-16

TABLE 4.3 (Continued)

COMPB1 TAPE OUTPUT FORMAT
RECORD TYPE 2

Data Item	Bits	Word
Record Type	4	1
Blanks	36	
ID	8	
Time	12	
Time	4	2
Sample number	8	
Value	8	
ID	8	
Time	16	
Sample number	8	
Value	8	
ID	8	3
Time	16	
Sample number	8	
Value	8	
ID	8	
Time	12	Repeat 7 times (15-16)
Time	4	4
Sample number	8	
Value	8	
ID	8	
Time	16	
Sample number	8	
Value	8	

TABLE 4.3 (Concluded)

COMPB1 TAPE OUTPUT FORMAT
RECORD TYPE 3

Data Item	Bits	Word
Record Type	4	1
Aircraft Title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Flight Duration (in seconds)	60	4
Blank	720	5-16

TABLE 4.4

COMPB2 TAPE OUTPUT FORMAT
RECORD TYPE 1

Data Item	Bits	Word
Record Type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Q window	8	4
R window	8	
Q-DOT Window	8	
R-DOT window	8	
NZ window	8	
NY window	8	
S2 window	8	
Blank	4	
S3 window	8	5
S6 window	8	
DWLR window	8	
S4 window	8	
S5 window	8	
DELE window	8	
VG window	8	
Blank	4	
DELF window	8	6

TABLE 4.4 (Continued)

COMPB2 TAPE OUTPUT FORMAT
RECORD TYPE 1

Data Item	Bits	Word
DELNG window	8	
PS window	8	
PT window	8	
DELP window	8	
Blank	20	
Q Cal 1	8	7
R Cal 1	8	
NZ Cal 1	8	
NY -(S1) Cal 1	8	
S2 Cal 1	8	
S3 Cal 1	8	
S6 Cal 1	8	
Blank	4	
DELR Cal 1	8	8
S4 Cal 1	8	
S5 Cal 1	8	
DELE Cal 1	8	
VG Cal 1	8	
DELF Cal 1	8	
DELNG Cal 1	8	
Blank	4	
PS Cal 1	8	9
PT Cal 1	8	
DELP Cal 1	8	
Q Cal 2	8	
R Cal 2	8	
NZ Cal 2	8	
NY (S1) Cal 2	8	
Blank	4	
S2 Cal 2	8	10

TABLE 4.4 (Continued)

COMPB2 TAPE OUTPUT FORMAT
RECORD TYPE 1

Data Item	Bits	Word
S3 Cal 2	8	
S6 Cal 2	8	
DELR Cal 2	8	
S4 Cal 2	8	
S5 Cal 2	8	
DELE Cal 2	8	
Blank	4	
VG Cal 2	8	11
DELF Cal 2	8	
DELNG Cal 2	8	
PS Cal 2	8	
PT Cal 2	8	
DELP Cal 2	8	
Q Cal 3	8	
Blank	4	
R Cal 3	8	12
NZ Cal 3	8	
NY (S1) Cal 3	8	
S2 Cal 3	8	
S3 Cal 3	8	
S6 Cal 3	8	
DELR Cal 3	8	
Blank	4	
S4 Cal 3	8	13
S5 Cal 3	8	
DELE Cal 3	8	
VG Cal 3	8	
DELF Cal 3	8	
DELNG Cal 3	8	
PS Cal 3	8	
Blank	4	
PT Cal 3	8	14
DELP Cal 3	8	
Blank	44	
Blank	120	15-16

TABLE 4.4 (Continued)

COMPB2 TAPE OUTPUT FORMAT
RECORD TYPE 2

Data Item	Bits	Word	
Record Type	4	1	
Blanks	36		
ID	8		
Time	12		
Time	4	2	
Sample number	8		
Value	8		
ID	8		
Time	16		
Sample number	8		
Value	8		
ID	8	3	
Time	16		
Sample number	8		
Value	8		
ID	8		
Time	12		
Time	4	4	Repeat 7 times (5-16)
Sample number	8		
Value	8		
ID	8		
Time	16		
Sample number	8		
Value	8		

TABLE 4.4 (Concluded)

COMPB2 TAPE OUTPUT FORMAT
RECORD TYPE 3

Data Item	Bits	(tail record) Word
Record type	4	1
Aircraft title	24	
Blank	32	
Blank	20	2
DD1	8	
DD2	8	
DD3	8	
DD4	8	
DD5	8	
DD6	8	3
DD7	8	
DD8	8	
DD9	8	
DD10	8	
DD11	8	
DD12	8	
Blank	4	
Flight Duration (in seconds)	60	4
Blank	720	5-16

SECTION 5
RESULTS

All four of the data compression programs were run extensively on RT data from the A-10, C-130, and C-141 aircraft. In order to establish the sensitivity of the Method A compression to the threshold size and the effect of the individual parameters on compression, runs were made with given values of the threshold on an individual parameter with the threshold on all other parameters set to a high value. Thus, the number of seconds containing data above threshold for each parameter was determined for three different threshold levels. The results of these runs for the three aircraft are shown in Tables 5.1, 5.2, and 5.3. It should be noted that the upper number in each block is the number of active seconds triggered by the specific parameter at the indicated threshold while the lower number is the number of times the flight condition or event changed when the parameter did not change. In all cases the windows on the flight condition parameters (altitude, airspeed, and fuel counter) were maintained at four counts. All runs were made with the option to process ground data in effect. After noting that the number of flight condition changes were small compared to the total number of active seconds, Programs COMPA1 and COMPA2 were modified to write Type 2 data records when these changes occurred. The programs were originally designed to write data records upon detecting a parameter outside threshold and a time slice record when a flight condition parameter changed. Elimination of the time slice records simplified program logic with only a small addition of data to the output tape.

The same type of sensitivity runs were made using Method B on the A-10 and C-141 data. Results of these runs are presented in Tables 5.4 and 5.5. The data contained in these tables are window crossings on individual parameters by flight and total peaks by parameter for all flights. Although it is unknown what practical use one might make of the information, it is interesting to note that the ratio of total peaks to total window crossings

TABLE 5.1

ACTIVE DATA BY PARAMETER USING METHOD A (A-10)

Flight No.	Flight Time Sec.	Threshold = 4					Threshold = 9					Threshold = 19				
		N_z	N_y	P	Q	R	N_z	N_y	P	Q	R	N_z	N_y	P	Q	R
1	7470	620 7	1263 4	179 7	692 6	1013 7	189 7	141 7	65 7	193 6	392 7	28 7	21 7	14 7	21 7	87 7
2	10380	3065 8	3525 4	1389 12	2509 6	2959 17	1651 10	921 14	467 16	1547 8	1279 21	909 11	132 19	75 20	736 12	119 21
3	6310	983 6	1595 6	131 6	569 5	1036 7	255 7	203 7	18 7	173 6	299 7	62 7	2 7	1 7	38 7	72 7
4	8590	3093 5	2544 5	1751 7	2977 3	2936 5	2032 5	1135 10	1030 10	1873 3	1493 6	1447 5	498 11	480 12	1294 7	392 11
5	8150	1667 10	2796 8	595 11	1617 9	2138 11	881 12	287 11	292 12	891 12	779 12	489 12	90 11	84 13	428 12	93 13
6	9090	2459 5	3142 2	1355 8	2273 3	2478 8	1525 6	1075 4	661 8	1408 6	1212 9	952 9	336 7	220 8	803 6	76 9
TOTAL	49990	11883 41	14865 29	5400 51	10637 32	12560 55	6533 47	3862 53	2533 60	6085 41	5454 62	3887 48	1079 62	874 67	3320 51	839 68

NOTE: Upper entry is seconds of active data.
Lower entry is number of flight condition changes.

TABLE 5.2

ACTIVE DATA BY PARAMETER USING METHOD A (C-151)

Flight No.	Flight Time Sec.	Threshold = 4				Threshold = 9				Threshold = 19			
		N_z	N_y	Q	R	N_z	N_y	Q	R	N_z	N_y	Q	R
1	19720	5871 10	22 12	90 8	366 11	31 12	1 12	22 12	209 11	13 12	1 12	1 12	80 11
2	32740	847 3	387 4	374 1	724 5	3 6	368 6	39 4	354 6	1 6	364 6	1 6	64 6
3	2150	193 1	3 2	6 1	50 2	3 2	2 2	4 1	22 2	2 2	2 2	2 2	11 2
TOTAL	54610	6911 14	412 18	470 10	1140 18	37 20	371 20	65 17	585 19	16 20	367 20	4 20	155 19

NOTE: Upper entry is seconds of active data.
Lower entry is number of flight condition changes.

TABLE 5.3

ACTIVE DATA BY PARAMETER USING METHOD A (C-130)

Flight No.	Flight Time Sec.	Threshold = 4			Threshold = 9			Threshold = 19		
		N_z	Q	R	N_z	Q	R	N_z	Q	R
1	16180	4468 26	2690 23	5720 41	879 49	192 63	3189 64	19 62	1 66	273 66
2	2940	625 2	224 2	433 5	117 4	14 4	155 6	16 6	1 6	62 6
3	18930	6129 24	3801 16	6000 29	1773 53	851 51	3615 59	109 62	13 64	427 64
4	6360	1524 8	956 8	1994 8	306 15	157 11	1025 14	8 16	8 16	160 16
5	10950	1960 10	930 14	3545 6	423 15	169 16	1482 14	64 17	38 16	121 17
TOTAL	55360	14726 70	8610 63	17692 89	3498 136	1383 145	9466 157	216 163	61 168	1042 169

NOTE: Upper entry is seconds of active data.
Lower entry is number of flight condition changes.

TABLE 5.4

A-10 METHOD B SAMPLES OF ACTIVE DATA

Flight No.	Flight Time Sec.	Window/Parameter														
		4NZ	4NY	4P	4Q	4R	9NZ	9NY	9P	9Q	9R	19NZ	19NY	19P	19Q	19R
1	7470	*	1137	367	704	638	175	181	143	165	220	47	46	68	47	67
2	10380		6245	2888	4139	1764	1952	1295	960	1243	530	639	318	247	466	161
3	6310	1848	2528	422	514	647	269	386	59	84	161	45	36	25	43	72
4	8590	5521	7284	5799	6715	3382	2512	2961	2587	2662	1277	974	1161	1041	1091	424
5	8150	2166	2269	1332	1635	1956	722	554	542	498	357	311	185	196	201	137
6	9090	4672	5128	2866	3367	1779	1682	1858	1169	1150	580	733	382	394	481	157
Total Active Data	49990	*	24591	13674	17074	9266	7312	6735	5460	5802	3125	2749	2128	1971	2329	1018
Total Peaks		*	8632	3390	5090	2820	1938	2242	1584	1720	1040	846	*	666	884	388
Peaks/ Total		*	.35	.25	.30	.30	.26	.33	.29	.30	.33	.31	*	.34	.38	.38

*Values not available due to tape break late in the program.

TABLE 5.5

C-141 METHOD B SAMPLES OF ACTIVE DATA

Flight	Time	Wind/Parameter											
		4NY	4NZ	4Q	4R	9NY	9NZ	9Q	9R	19NY	19NQ	19Q	19R
1	1970	28	22	21	104	21	21	21	62	21	21	21	40
2	19722	77	1567	137	195	41	92	45	103	41	44	41	69
3	32734	7951	650	1353	1816	5912	68	175	399	3670	31	31	53
Totals	54426	8056	2239	1511	2115	5964	181	241	564	3732	96	93	162
Total Peaks		1851	1107	705	891	1661	65	95	227	1445	19	19	51

does not vary from parameter to parameter for the A-10 data. This ratio also does not vary much as the window size is increased. It should be kept in mind that for Method A the data presented is seconds of active data and the data presented for Method B is samples of active data.

A direct comparison of the amount of output tape used for the two methods is not straight forward. It was first believed that by choosing the window sizes for Method B to be the same as the thresholds used in Method A, with the exception of the $\dot{p}$, $\dot{q}$, and $\dot{r}$ windows, a reasonable comparison might be made. Using thresholds and windows of 19 counts for $\dot{q}$ and $\dot{r}$ and 9 counts for all other parameters on C-141 data, the results shown in the first two columns of Table 5.6 were obtained. Since this seemed to be a large number of active samples using Method B, a run was made eliminating $\dot{q}$ and $\dot{r}$. The last column of Table 5.6 shows these results. If we assume that there will be a like number of window crossings and peaks on $\dot{q}$ and $\dot{r}$ to the number of peaks and window crossings on N_y, N_z, q and r we can linearly extrapolate for the total number of peaks and window crossings. This, of course, assumes equivalent window values on all parameters. Using the values from Table 5.5, with a 9 count window there are 1,800 windows and peaks per parameter for this data. This would give us 12,171 windows and peaks for all parameters on the tape. Since these values are blocked 23 per 16 word record on the tape, we would have a total of 8,464 words counting the header and end of flight records. The output from Method A is written as a 38-word record for each active second which gives a total of 46,322 words. Therefore, on this particular flight there is approximately 5.5 times more output for Method A than for Method B. It should be noted, however, that there appears to be an instrumentation problem with N_y on this tape. This can be seen by examination of Tables 5.2 and 5.5. The number of exceedances using Method A does not drop off significantly as the threshold is increased and the number of window crossings and peaks also does not decrease rapidly enough as the window value is increased. This leads one to believe that the actual ratio of output data for the two methods would be different for valid data. The CYBER 175

TABLE 5.6

COMPARISON OF METHODS FOR C-141 DATA

Flight No.	Time	Method A	Method B	
		Active Seconds[1]	Active Samples[2]	Active Samples[3]
1	48	1	23	21
2	1,970	118	434	430
3	19,722	274	9,299	781
4	32,734	601	39,761	7,297
5	92	1	113	21
6	142	1	22	21
Totals	54,708	996	49,652	8,571
CP Time[4] (Sec/Flt.Hour)		11.5	28.4	28.1

Notes:

(1) Threshold of 9 counts on all parameters.

(2) Window of 9 counts on all parameters except 19 counts/sec on $\dot{q}$ and $\dot{r}$.

(3) Window of 9 counts on all parameters except 999 counts/sec on $\dot{q}$ and $\dot{r}$.

(4) Central processor time on CYBER 175.

central processor time, however, is approximately 2.5 times greater for Method B than for Method A. This is due to the greater number of parameters examined for window crossings in Method B and in particular to the continuous computation of $\dot{p}$ and $\dot{r}$.

The same runs and extrapolation were made for the A-10 data. These runs were made on a shortened A-10 tape due to a stretching of the tape late in the program and the first two flights do not agree with previous runs. The results are shown in Table 5.7. The total number of windows and peaks were calculated to be 49,350. The total number of words output for Method B was calculated to be 34,704 and for Method A it was 386,700 which is about 11 times that of Method B. The amount of central processor time for the A-10 using Method B was only about 1.7 times that using Method A. This is due to the fact that there are not as many additional parameters analyzed for the A-10 using Method B over Method A.

The two runs made with realistic thresholds and windows for the A-10 tend to verify the calculation done above. The output of these runs was shown in Figures 4.1 and 4.2. Calculations made using this data show the compression ratio for Method A to be 2.36 and for Method B to be 26.9.

TABLE 5.7

COMPARISON OF METHOD FOR A-10 DATA

Flight Number		Time	Method A Active Samples	Method B Active Samples
1		14,832	4,428	7,771
2		6,318	1,496	1,001
3		8,170	3,581	12,523
9		8,152	2,312	3,128
10		9,090	3,330	7,130
	TOTALS	46,562	15,147	31,553
	CP Time (Sec/Flt. Hour)		13.9	23.5

SECTION 6
CONCLUSIONS AND RECOMMENDATIONS

The following conclusions and recommendations have been reached concerning the two data compression techniques:

1. Method A may be more suitable for fighter-type aircraft since more precise coincident values of parameters are obtained when any parameter peaks.

2. Method B can be used to obtain coincident values of other parameters if one can accept the degradation in accuracy due to using the previous window crossing on the coincident parameter.

3. Method B uses considerably less tape for output, thus a greater compression ratio is obtained. (The actual ratio of Method A to Method B is believed to be about 10 to 1.)

4. Method B uses significantly more computer time than Method A, but time used in subsequent analysis would be less. (This ratio will vary with the aircraft from about 1.5 to 2.5.)

5. The compression ratio for Method A, processing ground data, on the A-10 data using 11 counts on N_z, 13 counts on N_y, 5 counts on P, 6 counts on q and r, 9 counts on altitude, and 4 counts on airspeed and fuel counter was 2.36. This is the direct ratio of bits on the RT tape to bits on the output tape. The compression ratio using these same values as windows in Method B was 26.9.

6. The compression ratio using Method B would be considerably greater.

7. A study should be performed to examine the validity of the output data of the compression programs. This would require writing an edit program to select peaks and block the data.

8. Neither method of compression would affect peaks except for some data lost in the threshold needed to define a peak.

APPENDIX A

DATA COMPRESSION PROGRAM LISTINGS

APPENDIX A

DATA COMPRESSION PROGRAM LISTINGS

A complete source listing of each of the four data compression programs is provided in the following pages. Programs COMPA1, COMPA2, COMPB1, and COMPB2 are presented in Tables A.1 through A.4 while the general subroutines PACK and UNPACK are presented in Table A.5. In addition to the program listings, Table A.6 and A.7 contain definitions of variables used in the programs and Figures A.1 and A.2 show the general flow of Method A and Method B, respectively.

TABLE A.1

PROGRAM COMPA1


```
      PROGRAM COMPA1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE7)
C ************** RON JAHNING UDRI 547 CONTRACT# 93579 ************
C ****************************************************************
C  THIS PROGRAM IS THE FIRST COMPRESSION METHODOLOGY USING INITIALLY
C    DETERMINED THRESHOLDS AS DETERMINANTS AS TO WHETHER A SECOND
C   IS ACTIVE OR INACTIVE DATA.
C ******************************************************************
      DIMENSION NTBLK(324),ITBL(486)
      COMMON INDEX,IREC(486),IOVP,IOVR,IOVQ,IOVNY,IOVNZ,IOVALT,IOVAS,
     *IOVFC,IOVE1,IOVE5
      COMMON/CH1/ LAST
      COMMON/CH3/ IAC
      COMMON/CH4/ INCRST1,ITM
      COMMON/CNTR/ KOUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON/CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON/TCNTR/ KTKNT,LTKNT,NTKNT,KTNTR2
      COMMON/TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON/SHTCH/ ISHTCH1,ISHTCH2,ISHTCH3,IEND
      COMMON/TEMP/ ITEMP(6)
      COMMON/THRES/ ITP,ITQ,ITR,ITNY,ITNZ,ITALT,ITAS,ITFC
      DATA ITBL/486*0/
C
C  ********INITIALIZATION****
C       IOV=INTEGER OLD VALUE FOR...
C       CNT=COUNTERS FOR DIFFERENT AREAS...
C
      IOVP=IOVR=IOVQ=IOVNY=IOVNZ=IOVALT=IOVAS=IOVFC=IOVE1=IOVE5=0
      IERRCNT=ISHTCNT=IGRDCNT=KOUNT=LKOUNT=NKOUNT=IKOUNT=0
      ISHTCH2=ISHTCH4=KNTR1=KNTR2=KNTR3=0
      ITERCNT=ITSTCNT=ITGDCNT=KTNTR2=KTKNT=LTKNT=0
      ITKNT=NTKNT=0
      INFLGHT=ITM=IEND=0
      ISHTCH3=1
      ISHTCH1=INCRST1=2
      READ(5,1)IAC,NGR,ITNZ,ITNY,ITP,ITQ,ITR,ITALT,ITAS,
     *         ITFC,IOVNZ,IOVNY,IOVP,IOVQ,IOVR
    1 FORMAT(A4,I1,13I3)
      WRITE (6,7)IAC
    7 FORMAT(10H1AIRCRAFT:,2X,A4)
      IF(NGR.EQ.1)WRITE(6,8)
      IF(NGR.EQ.0)WRITE(6,9)
    8 FORMAT(23H PROCESSING GROUND DATA)
    9 FORMAT(27H NOT PROCESSING GROUND DATA)
      WRITE(6,11) ITNZ,ITNY,ITP,ITQ,ITR,ITALT,ITAS,ITFC
   11 FORMAT(43H THRESHOLDS: NZ- NY-  P-  Q-  R-ALT- AS- FC,
     */12X,8(I3,1X))
      WRITE(6,14)IOVNZ,IOVNY,IOVP,IOVQ,IOVR
   14 FORMAT(30H VALUES OF: NZ- NY-  P-  Q-  R,/11X,
     *5(I3,1X))
      LAST=0
   10 BUFFER IN(1,1)(NTBLK(1),NTBLK(324))
      IF (UNIT(1)) 20,900,15
   15 NKOUNT=NKOUNT+10
      GO TO 10
   20 IM=JM=1
      NKOUNT=NKOUNT+10
C
```

TABLE A.1 (Continued)

PROGRAM COMPA1


```
C     ********MAIN LOOP DECLARATION****
C          2 SECONDS/UNPACK...10 SECONDS/BUFFER IN.
C
        DO 80 K=1,5
          CALL UNPACK(IREC,486,NTBLK,IM,JM,ITBL)
          IF(ISWTCH4.EQ.0)GO TO 73
C
C    ********PARITY ERROR CHECK***********
C
          IF(OR(SHIFT(IREC(483),8),IREC(484)).NE.0) GO TO 71
C
C    ********SHORT RECORD CHECK***********
C
          IF(AND(IREC(480),15).NE.9) GO TO 72
          INDEX=0
          DO 70 L=1,2
            IF(L.EQ.2)INDEX=240
            JBIT=IREC(114+INDEX)/128
            JCAL=AND((IREC(114+INDEX)/64),1)
            JPRE=AND((IREC(114+INDEX)/32),1)
C
C       ********BIT CYCLE DETERMINATION**********
C
            IF(JBIT.EQ.1.AND.INFLGHT.NE.32.AND.IEND.NE.1)CALL PRT3
            IF(JBIT.EQ.1.OR.JCAL.EQ.1.OR.JPRE.EQ.1)IKOUNT=IKOUNT+1
            IF(JCAL.EQ.1.OR.JPRE.EQ.1)GO TO 70
            IF(JBIT.EQ.1)GO TO 35
C
C  CALCULATION OF THE SECONDS INTO FLIGHT
C
            ITM=NKOUNT-10+2*(K-1)+L
            IEND=0
C
C     *********IN FLIGHT CHECK**************
C
            INFLGHT=AND(IREC(16+INDEX),32)
            IF(INFLGHT.EQ.0)IGRDCNT=IGRDCNT+1
            IF(NGR.EQ.1)GO TO 25
            IF(INFLGHT.EQ.0)GO TO 70
C
C ****************************************************************
C          DONE ONLY ONCE AFTER BIT CYCLE HAS BEEN SHUT DOWN
C          INITIATION OF THE MEAN VALUES OF ALT,AS,FC,E(1-9)
C ****************************************************************
C
   25       IF(ISWTCH1.NE.0)GO TO 30
              ISWTCH1=2
            DO 26 M=6,22,16
              DO 26 N=1,3
                M1=M+N+INDEX
                IREC(M1)=128
   26       CONTINUE
            DO 27 M=38,198,16
              DO 27 N=1,3
                M1=M+N+INDEX
                IREC(M1)=1.25*((IREC(M1-27)-IREC(M1+37))+
    1                8*(IREC(M1+21)-IREC(M1-11)))+128
```

TABLE A.1 (Continued)

PROGRAM COMPA1


```
   27     CONTINUE
          ISWTCH2=0
          CALL PRT2
          IOVALT=IREC(98+INDEX)
          IOVAS=IREC(1+INDEX)
          IOVFC=IREC(6+INDEX)
          IOVE1=IREC(16+INDEX)
          IOVE5=IREC(17+INDEX)
   30     CALL ACTDATA
          GO TO 70
C
C *****************************************************
C STORE THE CALIBRATED DATA INTO THE TYPE1 RECORDS.
C *****************************************************
C
   35     IF(INFLGHT.NE.32)CALL PRT1
   70   CONTINUE
        GO TO 80
   71   IERRCNT=IERRCNT+2
        GO TO 80
   72   ISHTCNT=ISHTCNT+2
        GO TO 80
   73   ISWTCH4=2
   80 CONTINUE
      GO TO 10
C
C ******************************************************
C FINAL TAIL AND STATISTICAL EVALUATION PRINTOUT.
C ******************************************************
C
  900 IF(ISWTCH4.EQ.0)GO TO 10
      ISWTCH3=1
      CALL PRT3
      WRITE(6,1010)
 1010 FORMAT(32H1PRINTOUT FOR COMPA1 STATISTICS.)
      WRITE(6,1050)KNTR1
 1050 FORMAT(41H01.  THE NUMBER OF TYPE1 RECORDS PRINTED*,I7)
      WRITE(6,1060)KTNTR2
 1060 FORMAT(41H02.  THE NUMBER OF TYPE2 RECORDS PRINTED*,I7)
      WRITE(6,1070)KNTR3
 1070 FORMAT(41H03.  THE NUMBER OF TYPE3 RECORDS PRINTED*,I7)
      WRITE(6,1080)ITGDCNT
 1080 FORMAT(28H04.  SECONDS OF GROUND DATA*,I7)
      WRITE(6,1085)KTKNT
 1085 FORMAT(26H05.  TOTAL ACTIVE SECONDS*,I7)
      WRITE(6,1095)NTKNT
 1095 FORMAT(29H06.  TOTAL NUMBER OF SECONDS*,I7)
      WRITE(6,1100)
 1100 FORMAT(4H EOJ)
      STOP
      END
```

TABLE A.1 (Continued)
PROGRAM COMPA1


```
      SUBROUTINE ACTDATA
      DIMENSION LASTVAL(12)
      COMMON INDEX,IREC(486),IOVP,IOVR,IOVQ,IOVNY,IOVNZ,IOVALT,IOVAS,
     *IOVFC,IOVE1,IOVE5
      COMMON/CM1/ LAST
      COMMON/CM4/ INCRST1,ITM
      COMMON/CNTR/ KOUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON/CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON/TCNTR/ KTKNT,LTKNT,NTKNT,KTNTR2
      COMMON/TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON/SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      COMMON/TEMP/ ITEMP(6)
      COMMON/THRES/ ITP,ITQ,ITR,ITNY,ITNZ,ITALT,ITAS,ITFC
C
C ****************************************************************
C*******ACTIVE DATA-DETERMINED BY IF OLD VALUE + THESHOLD<VALUE<
C     OLD VALUE - THESHOLD
C     VALUES ARE-NY,NZ,P,Q,R,ALT,AS,E1,E5,FCTR
C     OLD VALUES DEFINED PREVIOUSLY
C     THRESHOLDS ARE INPUT
C ****************************************************************
C
      N1=6
      DO 10 M=1,225,16
         M1=M+9+INDEX
      IF(IABS(IREC(M1)-IOVNY).GT.ITNY.OR.
     1   IABS(IREC(M1+2)-IOVR).GT.ITR.OR.
     2   IABS(IREC(M1+3)-IOVP).GT.ITP.OR.
     3   IABS(IREC(M1+4)-IOVQ).GT.ITQ.OR.
     4   IABS(IREC(M1+5)-IOVNZ).GT.ITNZ) GO TO 20
   10 CONTINUE
      IF(IABS(IREC(98+INDEX)-IOVALT).GT.ITALT.OR.
     1 IABS(IREC(1+INDEX)-IOVAS).GT.ITAS.OR.
     2 IOVE1.NE.IREC(16+INDEX).OR.
     3 IOVE5.NE.IREC(17+INDEX))GO TO 20
      DO 15 N1=6,230,16
        IF(IABS(IREC(N1+INDEX)-IOVFC).GT.ITFC)GO TO 20
   15 CONTINUE
C
C ****************************************************************
C   INACTIVE DATA RECOGNIZED.
C   ****LAST WILL BE SET TO ONE IF PREVIOUS DATA WAS ACTIVE  ..THIS WILL
C       ALLOW FOR EACH ACTIVE DATA TO BE FOLLOWED BY ONE SECOND
C ****************************************************************
      LKOUNT=LKOUNT+1
      IF(LAST.EQ.1)GO TO 50
      RETURN
   20 IF(KOUNT.NE.0)GO TO 25
      ISWICH1=1
      CALL PRT1
      KOUNT=KOUNT+1
      ISWTCH2=1
      DO 23 M=1,6
        ITEMP(M)=120
   23 CONTINUE
      CALL PRT2
   25 IOVALT=IREC(98+INDEX)
```

TABLE A.1 (Continued)

PROGRAM COMPA1


```
      IOVA3=IREC(1+INDEX)
      IOVFC=IREC(N1+INDEX)
      IOVE1=IREC(16+INDEX)
      IOVE5=IREC(17+INDEX)
      INCR=0
      KOUNT=KOUNT+1
C
C ***************************************************************
C********RESET THE OLD VALUES ACTIVE DATA FOUND.....
C          SAVE LAST FOUR VALUES OF P,Q,R,(LASTVAL)
C          THEN CALCULATE P.-Q.-R. FOR 3-13
C          SET 1-2 P.-Q.-R. TO 128.
C ***************************************************************
C
      DO 30 M=187,235,16
         DO 30 N=1,3
            INCR=INCR+1
            LASTVAL(INCR)=IREC(M+N+INDEX)
   30 CONTINUE
      IF(LAST.EQ.1)GO TO 110
      DO 35 M=6,22,16
         DO 35 N=1,3
            M1=M+N
            IREC(M1)=128
   35 CONTINUE
      DO 40 M=38,198,16
         DO 40 N=1,3
            M1=M+N+INDEX
            IREC(M1)=(1.25)*((IREC(M1-27)-IREC(M1+37))+
     1                  8*(IREC(M1+21)-IREC(M1-11)))+128
   40 CONTINUE
      LAST=1
      CALL PRT2
      RETURN
C
C ***************************************************************
C********ACTIVE DATA NOT FOUND CALCULATE LAST 2 P.-Q.-R. FOR 14-15 OF PR
C     RECORD(ITEMP(1-6)).  STORE INTO TYPE 2, GO ON AND CALCULATE P.-Q.-
C       THIS RECORD AND STORE INTO TYPE 2.....********
C ***************************************************************
C
   50 LAST=0
      DO 70 M=1,3
         ITEMP(M)=(1.25)*((LASTVAL(M)-IREG(M+11+INDEX))+
     1                8*(LASTVAL(M+9)-LASTVAL(M+3)))+128
         ITEMP(M+3)=(1.25)*((LASTVAL(M+3)-IREG(M+27+INDEX))+
     1               8*(IREC(M+11+INDEX)-LASTVAL(M+6)))+128
   70 CONTINUE
      ISWTCH2=1
      CALL PRT2
      DO 80 M=1,3
         IREC(6+M+INDEX)=(1.25)*((LASTVAL(M+6)-IREC(M+43+INDEX))+
     1             8*(IREC(M+27+INDEX)-LASTVAL(M+9)))+128
   80 CONTINUE
      DO 90 M=1,3
         IREG(22+M+INDEX)=(1.25)*((LASTVAL(M+9)-IREC(M+59+INDEX))+
     1             8*(IREC(M+43+INDEX)-IREC(M+11+INDEX)))+128
```

TABLE A.1 (Continued)

PROGRAM COMPA1


```
   90 CONTINUE
      DO 100 M=38,198,16
           DO 100 N=1,3
           M1=M+N+INDEX
           IREC(M1)=(1.25)*((IREC(M1-27)-IREC(M1+37))+
     1                        8*(IREC(M1+21)-IREC(M1-11)))+128
  100 CONTINUE
      CALL PRT2
      ISWTCH2=1
      DO 105 M=1,6
         ITEMP(M)=128
  105 CONTINUE
      CALL PRT2
      IOVALT=IREC(98+INDEX)
      IOVAS=IREC(1+INDEX)
      IOVFC=IREC(N1+INDEX)
      IOVE1=IREC(16+INDEX)
      IOVE5=IREC(17+INDEX)
      RETURN
C
C ******************************************************************
C*********ACTIVE DATA FOUND AND FOLLOWS ANOTHER ACTIVE DATA//
C        CALCULATE LAST 2 VALUES OF P.-Q.-R. FOR 14-15
C          PREVIOUS RECORD (ITEMP(1-6)).  STORE INTO TYPE 2,GO ON AND
C        CALCULATE P.-Q.-R. FOR 1-13. NOT TO BE STORED UNTIL 14-15
C        ARE CALCULATED.
C ******************************************************************
C
  110 DO 130 M=1,3
         ITEMP(M)=(1.25)*((LASTVAL(M)-IREC(M+11+INDEX))+
     1                         8*(LASTVAL(M+9)-LASTVAL(M+3)))+128
         ITEMP(M+3)=(1.25)*((LASTVAL(M+3)-IREC(M+27+INDEX))+
     1                        8*(IREC(M+11+INDEX)-LASTVAL(M+6)))+128
  130 CONTINUE
      ISWTCH2=1
      CALL PRT2
      DO 150 M=1,3
         IREC(6+M+INDEX)=(1.25)*((LASTVAL(M+6)-IREC(M+43+INDEX))+
     1                        8*(IREC(M+27+INDEX)-LASTVAL(M+9)))+128
         IREC(22+M+INDEX)=(1.25)*((LASTVAL(M+9)-IREC(M+59+INDEX))+
     1                    8*(IREC(M+43+INDEX)-IREC(M+11+INDEX)))+128
  150 CONTINUE
      DO 160 M=38,198,16
         DO 160 N=1,3
           M1=M+N+INDEX
           IREC(M1)=(1.25)*((IREC(M1-27)-IREC(M1+37))+
     1                         8*(IREC(M1+21)-IREC(M1-11)))+128
  160 CONTINUE
      CALL PRT2
      RETURN
      END
```

TABLE A.1 (Continued)

PROGRAM COMPA1

```
      SUBROUTINE PRI1         74/74    OPT=1 TRACE              FTN 4.7+476        10/19/79  16.41.39        PAGE     1

            SUBROUTINE PRI1
            DIMENSION IP1(25),IRECO1(64),ITBL1(64),LIST(11)
            COMMON INDEX,IREC(486)
            COMMON/CNTR/ KOUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
            COMMON/CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
            COMMON/ICNTR/ KTKNT,LTKNT,NTKNT,KTNTR2
            COMMON/ICNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
            COMMON/CM3/ IAC
            COMMON/CM4/ INCRST1,ITM
            COMMON/SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
            COMMON/THRES/ IIP,ITQ,ITR,ITNY,ITNZ,ITAL1,ITAS,ITFC
            DATA ITBL1/4,-24,52,12*0,4,7*0,4,7*0,4,7*0,4,7*0,4,7*0,4,7*0,4/
            DATA LIST /10,15,13,14,12,5,11,19,1,98,6/
      C
      C********REPEAT BEFORE EACH FLIGHT...IE.FIRST RECORD****
      C  *************SWITCH1 DEFINITION********
      C      ISWTCH1=0 WHEN IT HAS GONE THROUGH PREVIOUS CAL.
      C      ISWTCH1=1 WHEN IT HAS FOUND THE FIRST GOOD SEG.
      C                OF DATA---PRINT TYPE 1 RECORD.
      C      ISWTCH1=2 WHEN IT IS TO PROCESS CAL DATA THIS
      C                ALLOWS FOR INITIALIZATION INTO ARRAY.
      C  *************************************
            IF(ISWTCH1.EQ.1)GO TO 135
            IF(ISWTCH1.EQ.2) INCRST1=26
      C
      C   ****CALIBRATED INFORMATION****
      C
            INCR=INCRST1-1
            DO 30 M=1,11
               INCR=INCR+1
               IF(MOD(INCR,8).NE.0)GO TO 20
               IRECO1(INCR)=0
               INCR=INCR+1
         20    IRECO1(INCR)=IREC(LIST(M)+INDEX)
         30 CONTINUE
            ISWTCH1=0
            IF(INCR.GE.62)ISWTCH1=2
            INCRST1=INCR
            RETURN
        135 INCR=INCRST1-1
      C
      C *************CREATE TYPE3 RECORD*************************
      C
            ISWTCH3=0
            CALL PRT3
            DO 137 M=INCR,64
               IRECO1(M)=0
        137 CONTINUE
            IRECO1(1)=1
            IRECO1(2)=IAC
            IRECO1(3)=0
      C
      C   *****DOCUMENTARY DATA*****
      C
            INCR=4
            DO 140 M=63,127,16
               DO 140 I=1,3
```

TABLE A.1 (Continued)

PROGRAM COMPA1


```
      IF(M+N.EQ.98.OR.M+N.GT.128)GO TO 140
      IRECO1(INCR)=IREC(M+N+INDEX)
      INCR=INCR+1
  140 CONTINUE
      IRECO1(16)=0
C
C   ****THRESHOLDS****
C
      IRECO1(17)=ITALT
      IRECO1(18)=ITAS
      IRECO1(19)=ITFC
      IRECO1(20)=ITHY
      IRECO1(21)=ITNZ
      IRECO1(22)=ITP
      IRECO1(23)=ITQ
      IRECO1(24)=0
      IRECO1(25)=ITR
      IJ=IK=1
      DO 155 M=1,25
         IP1(M)=0
  155 CONTINUE
      CALL PACK(IRECO1,64,IP1,IJ,IK,ITBL1)
      ISWTCH1=2
      KNTR1=KNTR1+1
C     WRITE(7)(IP1(J),J=1,25)
      RETURN
      END
```

TABLE A.1 (Continued)

PROGRAM COMPA1

```
SUBROUTINE PRT2        74/74    OPT=1 TRACE                    FTN 4.7+476          10/19/79  16.41.39          PAGE    1

                  SUBROUTINE PRT2
                  DIMENSION IP2(25),IRECO2(187),ITBL2(187)
                  COMMON INDEX,IREC(486)
                  COMMON/CNTR/ KOUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
                  COMMON/CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
                  COMMON/TCNTR/ KTKNT,LTKNT,NTKNT,KTNTR2
                  COMMON/TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
                  COMMON/CN4/ INCRST1,ITM
                  COMMON/SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
                  COMMON/TEMP/ ITEMP(6)
                  DATA ITBL2/4,16,185*8/
              C
              C********DATA IN ACTIVE PERIODS****
              C********OUTPUTTED IF ACTIVE CONDITIONS ARE MET,OUTPUT 1 SEG AFTER GRITE
              C          FAILS****
                  IF(ISWTCH2.EQ.1)GO TO 20
              C
              C  ****TITLE****
              C  *****TYPE,TIME SEC.,AIR/SPEED,ALT,FCT,E(1-4),E(5-9)*******
              C
                  IRECO2(1)=2
                  IRECO2(2)=ITM
                  IRECO2(3)=IREC(1+INDEX)
                  IRECO2(4)=IREC(90+INDEX)
                  IRECO2(5)=IREC(6+INDEX)
                  IRECO2(6)=IREC(16+INDEX)
                  IRECO2(7)=IREC(17+INDEX)
              C
              C  ****NY,NZ,P,Q,R,P.,Q.,R.,SB,PLA******
              C
                  INCR=8
                  DO 10 M=3,227,16
                    IRECO2(INCR)=IREC(7+M+INDEX)
                    IRECO2(INCR+1)=IREC(12+M+INDEX)
                    IRECO2(INCR+2)=IREC(10+M+INDEX)
                    IRECO2(INCR+3)=IREC(11+M+INDEX)
                    IRECO2(INCR+4)=IREC(9+M+INDEX)
                    IRECO2(INCR+5)=IREC(5+M+INDEX)
                    IRECO2(INCR+6)=IREC(6+M+INDEX)
                    IRECO2(INCR+7)=IREC(4+M+INDEX)
                    IRECO2(INCR+8)=IREC(2+M+INDEX)
                    IRECO2(INCR+9)=IREC(M+INDEX)
                    IRECO2(INCR+10)=IREC(11+M+INDEX)
                    IRECO2(INCR+11)=0
                    INCR=INCR+12
                10 CONTINUE
                  RETURN
                20 INCR=0
                  ISWTCH2=0
                  KNTR2=KNTR2+1
                  DO 30 M=168,180,12
                    DO 30 N=1,3
                      INCR=INCR+1
                      IRECO2(M+N)=ITEMP(INCR)
                30 CONTINUE
                  IK=IJ=1
                  DO 40 M=1,25
```

TABLE A.1 (Continued)

PROGRAM COMPA1

```
SUBROUTINE PRT2        74/74   OPT=1 TRACE                    FTN 4.7+476        10/19/79  16.41 39        PAGE     2

                    IP2(M)=0
                 40 CONTINUE
                    CALL PACK(IRECO2,187,IP2,IJ,IK,ITBL2)
              C     WRITE(7) (IP2(J),J=1,25)
                    RETURN
                    END
```

TABLE A.1 (Continued)

PROGRAM COMPA1


```
      SUBROUTINE PRT3
      DIMENSION IP3(25),IRECO3(17),ITBL3(17)
      COMMON INDEX,IREC(486)
      COMMON/CH3/ IAC
      COMMON/CH4/ INCRST1,ITM
      COMMON/CNTR/ KCUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON/CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON/TCNTR/ KTKNT,LTKNT,NTKNT,KTNTR2
      COMMON/TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON/SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      DATA ITBL3/4,-24,52,12*8,4,60/
C
C*********END OF FLIGHT (TAIL)****
C *********SWITCH3 DEFINTION*******
C       ISWTCH3=0 WHEN INITIAL ENTRY STOR DOC DATA.
C       ISWTCH3=1 WHEN ITS TO SAVE TIME SEC. AND WRITE TYPE 3 REC.
C ***********************************
      IF (ISWTCH3.EQ.1)GO TO 11
      IRECO3(1)=3
      IRECO3(2)=IAC
      IRECO3(3)=0
C
C*********DOCUMENTARY DATA****
C
      INCR=4
      DO 10 M=63,127,16
         DO 10 N=1,3
            IF(M+N.EQ.98.OR.M+N.GT.128)GO TO 10
            IRECO3(INCR)=IREC(M+N+INDEX)
            INCR=INCR+1
   10 CONTINUE
      ISWTCH3=1
      RETURN
   11 IRECO3(16)=0
      ISWTCH3=0
      IEND=1
      IJ=IK=1
      IRECO3(17)=ITM
      IF(KOUNT.EQ.0)GO TO 30
      DO 20 M=1,25
      IP3(M)=0
   20 CONTINUE
      CALL PACK(IRECO3,17,IP3,IJ,IK,ITBL3)
      KNTR3=KNTR3+1
C     WRITE(7)(IP3(J),J=1,25)
C
C ************************************************************
C FINAL INDIVIDUAL FLIGHT STATISTICS
C ************************************************************
C
      WRITE(6,1010)KNTR1
 1010 FORMAT(8H0FLIGHT#,I3,11H STATISTICS)
      WRITE(6,1020)KNTR2
 1020 FORMAT(41H 1.  THE NUMBER OF TYPE2 RECORDS PRINTED=,I7)
      WRITE(6,1060)IGRDCNT
 1060 FORMAT(28H 2.  SECONDS OF GROUND DATA=,I7)
      WRITE(5,1090)KCUNT
```

TABLE A.1 (Concluded)

PROGRAM COMPA1


```
 1090 FORMAT(28H 3.  SECONDS OF ACTIVE DATA*,I7)
      WRITE(6,1110)ITM
 1110 FORMAT(29H 4.  TOTAL NUMBER OF SECONDS*,I7)
   30 KTNTR2=KNTR2+KTNTR2
      ITERCNT=IERRCNT+ITERCNT
      ITSTCNT=ISHTCNT+ITSTCNT
      ITGDCNT=IGRDCNT+ITGDCNT
      ITKNT=IKOUNT+ITKNT
      KTKNT=KOUNT+KTKNT
      LTKNT=LKOUNT+LTKNT
      NTKNT=ITM+NTKNT
      KNTR2=IERRCNT=ISHTCNT=IGRDCNT=0
      IKOUNT=KOUNT=LKOUNT=0
      NKOUNT=NKOUNT-ITM
      RETURN
      END
```

TABLE A.2

PROGRAM COMPA2


```
      PROGRAM COMPA2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE7)
C     ******************RON JANNING  UDRI 547 53579-00*********************
C     *****************************************************************
C     THIS ROUTINE IS THE FIRST COMPRESSION METHODOLOGY DESIGNED
C     FOR THE C141 AND C130 MODELS.  USING INITIALLY DETERMINED
C     THRESHOLDS AS DETERMINANTS AS TO WHETHER A SECOND IS ACTIVE
C     OR INACTIVE DATA.
C     *****************************************************************
C
      DIMENSION NTBLK(324),ITBL(486)
      COMMON INDEX,IREC(486)
      COMMON /OLDVL/ IOVR,IOVQ,IOVNY,IOVNZ,IOVALT,IOVAS,IOVE1,IOVE5
      COMMON /CM1/ LAST
      COMMON /CM3/ IAC
      COMMON /CM4/ ITH
      COMMON /CNTR/ KOUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON /CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON /TCNTR/ KTKNT,LTKNT,NTKNT,KTNTR2
      COMMON /TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON /SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      COMMON /TEMP/ IQDOT(20),IRDOT(20)
      COMMON /THRES/ ITQ,ITR,ITNY,ITNZ,ITALT,ITAS
      DATA ITBL/486*0/
C
C     ************ INITIALIZATION ***************
C        IOV= INTEGER OLD VALUE FOR ....
C        CNT= COUNTERS FOR AREA ...
C
      IOVQ=IOVR=IOVNY=IOVNZ=IOVALT=IOVAS=IOVE1=IOVE5=0
      IERRCNT=ISHTCNT=IGRDCNT=KOUNT=LKOUNT=NKOUNT=IKOUNT=0
      ISWTCH2=ISWTCH4=KNTR1=KNTR2=KNTR3=0
      ITERCNT=ITSTCNT=ITGDCNT=KTNTR2=KTKNT=LTKNT=0
      INFLGHT=ITKNT=NTKNT=ITH=IEND=0
      ISWTCH3=1
      ISWTCH1=2
      READ(5,1)IAC,NGR,ITNZ,ITNY,ITQ,ITR,ITALT,ITAS,IOVNZ,IOVNY,IOVQ,
     *IOVR
    1 FORMAT(A4,I1,10I3)
      WRITE(6,4)IAC
    4 FORMAT(10H1AIRCRAFT:,4X,A4)
      IF(NGR.EQ.1)WRITE(6,5)
      IF(NGR.EQ.0)WRITE(6,6)
    5 FORMAT(23H PROCESSING GROUND DATA)
    6 FORMAT(27H NOT PROCESSING GROUND DATA)
      WRITE(6,9)ITNZ,ITNY,ITQ,ITR,ITALT,ITAS
    9 FORMAT(35H THRESHOLDS: NZ- NY-  Q-  R-ALT- AS,
     */12X,6(I3,1X))
      WRITE(6,14)IOVNZ,IOVNY,IOVQ,IOVR
   14 FORMAT(26H VALUES OF: NZ- NY-  Q-  R,/11X,4(I3,1X))
      LAST=0
   10 BUFFER IN(1,1)(NTBLK(1),NTBLK(324))
      IF(UNIT(1)) 20,900,15
   15 NKOUNT=NKOUNT+10
      GO TO 10
   20 IN=JN=1
      NKOUNT=NKOUNT+10
C
```

TABLE A.2 (Continued)

PROGRAM COMPA2


```
      C  *************** MAIN LOOP DECLARATION ******************
      C     2 SECONDS/UNPACK...10 SECONDS/BUFFER IN.
      C
            DO 100 K=1,5
               CALL UNPACK(IREC,486,NTBLK,IM,JM,ITBL)
               IF(ISHTCH4.EQ.0)GO TO 93
      C
      C  *************** PARITY ERROR CHECK ***********************
      C
               IF(OR(SHIFT(IREC(483),8),IREC(484)).NE.0) GO TO 91
      C
      C  *************** SHORT RECORD CHECK ***********************
      C
               IF(AND(IREC(480),15).NE.9) GO TO 92
               INDEX=0
               DO 90 L=1,2
                  IF(L.EQ.2) INDEX=240
      C
      C  *************** BIT CYCLE DETERMINATION *****************
      C
                  JBIT=IREC(232+INDEX)/128
                  JCAL=AND((IREC(232+INDEX)/64),1)
                  JPRE=AND((IREC(232+INDEX)/32),1)
                  IF(JBIT.EQ.1.AND.INFLGHT.NE.64.AND.IEND.NE.1)CALL PRT3
                  IF(JBIT.EQ.1.OR.JCAL.EQ.1.OR.JPRE.EQ.1)IKOUNT=IKOUNT+1
                  IF(JCAL.EQ.1.OR.JPRE.EQ.1)GO TO 90
                  IF(JBIT.EQ.1)GO TO 35
      C
      C CALCULATION OF THE SECONDS INTO THE FLIGHT
      C
                  ITM=NKOUNT-10+2*(K-1)+L
      C
      C    **********IN FLIGHT CHECK******************
      C
                  IEND=0
                  INFLGHT=AND(IREC(64+INDEX),64)
                  IF(INFLGHT.EQ.0)IGRDCNT=IGRDCNT+1
                  IF(NGR.EQ.1)GO TO 25
                  IF(INFLGHT.EQ.0)GO TO 90
      C
      C  ************************************************************
      C  DONE ONLY ONCE AFTER BIT CYCLE TEST HAS BEEN SHUT DOWN
      C  TO INITIALIZE THE MEAN VALUES OF THE ALT,AS,E(1-9)
      C  ************************************************************
      C
   25             IF(ISHTCH1.NE.0)GO TO 30
                     ISHTCH1=2
                  DO 26 M=1,2
                     IQOUT(M)=128
                     IROUT(M)=128
   26             CONTINUE
                  DO 27 M=3,18
                     IQOUT(M)=1.6667*((IREC((M-3)*12+1+INDEX)-
     1                           IREC((M+2)*12+1+INDEX))
     2                       +8*(IREC((M+1)*12+1+INDEX)-
     3                           IREC((M-1)*12+1+INDEX)))+128
                     IROUT(M)=1.6667*((IREC((M-3)*12+2+INDEX)-
```

TABLE A.2 (Continued)

PROGRAM COMPA2

```
      PROGRAM COMPA2     74/74   OPT=1 TRACE              FTN 4.7+476        11/13/79  17.17.23       PAGE    3

     1                            IREC((M+2)*12+2+INDEX))
     2                        +8*(IREC((M+1)*12+2+INDEX)-
     3                            IREC((M-1)*12+2+INDEX)))+128
   27       CONTINUE
            ISWTCH2=0
            CALL PRT2
            IOVALT=IREC(40+INDEX)
            IOVAS=IREC(47+INDEX)
            IOVE1=IREC(64+INDEX)
            IOVE5=IREC(112+INDEX)
   30       CALL ACTDATA
            GO TO 90
C
C  ************************************************************
C  THE CALIBRATED DATA INTO THE TYPE1 RECORDS.
C  ************************************************************
C
   35       IF(INELGHT.NE.64)CALL PRT1
   90     CONTINUE
          GO TO 100
   91     IERRCNT=IERRCNT+2
          GO TO 100
   92     ISHTCNT=ISHTCNT+2
          GO TO 100
   93     ISWTCH4=2
  100 CONTINUE
      GO TO 10
C  ************************************************************
C  TAPE MARK CAUSES END OF FILE IN THE BUFFER STATEMENT
C  SWITCH 4 IS BEING USED THEREFORE TO CATCH THIS ERROR
C  ************************************************************
  900 IF(ISWTCH4.EQ.0)GO TO 10
      ISWTCH3=1
      CALL PRT3
      WRITE(6,1010)
 1010 FORMAT(32H1PRINTOUT FOR COMPA2 STATISTICS.)
      WRITE(6,1050)KNTR1
 1050 FORMAT(41H01.  THE NUMBER OF TYPE1 RECORDS PRINTED=,I7)
      WRITE(6,1060)KTNTR2
 1060 FORMAT(41H02.  THE NUMBER OF TYPE2 RECORDS PRINTED=,I7)
      WRITE(6,1070)KNTR3
 1070 FORMAT(41H03.  THE NUMBER OF TYPE3 RECORDS PRINTED=,I7)
      WRITE(6,1080)ITGDCNT
 1080 FORMAT(28H04.  SECONDS OF GROUND DATA=,I7)
      WRITE(6,1090)KTKNT
 1090 FORMAT(26H05.  TOTAL ACTIVE SECONDS=,I7)
      WRITE(6,1130)NTKNT
 1130 FORMAT(29H06.  TOTAL NUMBER OF SECONDS=,I7)
      STOP
      END
```

TABLE A.2 (Continued)

PROGRAM COMPA2


```
      SUBROUTINE ACTDATA
      DIMENSION LASTQ(4),LASTR(4)
      COMMON INDEX,IREC(486)
      COMMON /OLDVL/ IOVR,IOVQ,IOVNY,IOVNZ,IOVALT,IOVAS,IOVE1,IOVE5
      COMMON /CM1/ LAST
      COMMON /CM4/ ITM
      COMMON /CNTR/ KOUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON /CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON /TCNTR/ KTKNT,LTKNT,NTKNT,KTKNTR2
      COMMON /TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON /SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      COMMON /TEMP/ IQDOT(20),IRDOT(20)
      COMMON /THRES/ ITQ,ITR,ITNY,ITNZ,ITALT,ITAS
C
C **************************************************************
C ACTIVE DATA: DETERMINED BY IF OLD VALUE +THRESHOLD <
C VALUE OR IF OLD VALUE -THRESHOLD > VALUE
C VALUES ARE: NY,NZ,Q,R
C OLD VALUES DEFINED FROM THE MEAN OF THE DISTRIBUTION
C THRESHOLD ARE INITIALIZED FROM THE INPUT
C **************************************************************
C
      DO 10 M=1,229,12
         IF(IABS(IREC(M+INDEX)-IOVQ).GT.ITQ.OR.
     #        IABS(IREC(M+1+INDEX)-IOVR).GT.ITR)GO TO 20
   10 CONTINUE
      DO 15 M=3,235,8
         IF(IABS(IREC(M+INDEX)-IOVNZ).GT.ITNZ.OR.
     #        IABS(IREC(M+1+INDEX)-IOVNY).GT.ITNY)GO TO 20
   15 CONTINUE
      IF(IABS(IREC(40+INDEX)-IOVALT).GT.ITALT.OR.
     1   IABS(IREC(47+INDEX)-IOVAS).GT.ITAS.OR.
     2   IOVE1.NE.IREC(64+INDEX).OR.
     3   IOVE5.NE.IREC(112+INDEX))GO TO 20
C
C **************************************************************
C INACTIVE DATA RECOGNIZED: LAST WILL BE SET TO ONE IF
C PREVIOUS DATA ACTIVE... THIS WILL ALLOW FOR EACH ACT
C IVE DATA TO BE FOLLOWED BY ONE SECOND OF INACTIVE
C DATA.
C **************************************************************
C
      LKOUNT=LKOUNT+1
      IF(LAST.EQ.1)GO TO 50
      RETURN
   20 IF(KOUNT.NE.0)GO TO 25
      ISWTCH1=1
      CALL PRT1
      KOUNT=KOUNT+1
      ISWTCH2=1
      DO 23 M=19,20
        IQDOT(M)=128
        IRDOT(M)=128
   23 CONTINUE
      CALL PRT2
   25 IOVALT=IREC(40+INDEX)
      IOVAS=IREC(47+INDEX)
```

TABLE A.2 (Continued)

PROGRAM COMPA2


```
          IOVE1=IREC(64+INDEX)
          IOVE5=IREC(112+INDEX)
          KOUNT=KOUNT+1
C
C *****************************************************************
C RESET THE OLD VALUES, ACTIVE DATA HAS BEEN FOUND..
C SAVE THE LAST FOUR VALUES OF Q,R..
C THEN CALCULATE Q.-R. FOR 3-18.
C SET 1-2 Q.-R. TO 128
C *****************************************************************
C
          INCR=0
          DO 30 M=193,229,12
             INCR=INCR+1
             LASTQ(INCR)=IREC(M+INDEX)
             LASTR(INCR)=IREC(M+1+INDEX)
       30 CONTINUE
          IF(LAST.EQ.1)GO TO 110
          DO 35 M=1,2
             IQDOT(M)=128
             IRDOT(M)=128
       35 CONTINUE
          DO 40 M=3,18
             IQDOT(M)=(1.6667)*((IREC((M-3)*12+1+INDEX)-IREC((M+2)*12+1+INDE
         #     X))+8*(IREC((M+1)*12+1+INDEX)-IREC((M-1)*12+1+INDEX)))+128
             IRDOT(M)=(1.6667)*((IREC((M-3)*12+2+INDEX)-IREC((M+2)*12+2+INDE
         #     X))+8*(IREC((M+1)*12+2+INDEX)-IREC((M-1)*12+2+INDEX)))+128
       40 CONTINUE
          LAST=1
          CALL PRT2
          RETURN
C
C *****************************************************************
C ACTIVE DATA NOT FOUND CALCULATE LAST 2 Q.-R. FOR 19-20
C OF PREVIOUS RECORD.  STORE INTO TYPE2, GO ON AND CALC-
C ULATE Q.-R. FOR THIS RECORD 1-18 AND STORE INTO TYPE2.
C *****************************************************************
C
       50 LAST=0
          IQDOT(19)=(1.6667)*((LASTQ(1)-IREC(1+INDEX))+8*(LASTQ(4)-LASTQ(2))
         #           )+128
          IQDOT(20)=(1.6667)*((LASTQ(2)-IREC(13+INDEX))+8*(IREC(1+INDEX)-LAS
         #           TQ(3)))+128
          IRDOT(19)=(1.6667)*((LASTR(1)-IREC(2+INDEX))+8*(LASTR(4)-LASTR(2))
         #           )+128
          IRDOT(20)=(1.6667)*((LASTR(2)-IREC(14+INDEX))+8*(IREC(2+INDEX)-LAS
         #           TR(3)))+128
          ISWTCH2=1
          CALL PRT2
          IQDOT(1)=(1.6667)*((LASTQ(3)-IREC(25+INDEX))+8*(IREC(13+INDEX)-LAS
         #          TQ(4)))+128
          IQDOT(2)=(1.6667)*((LASTQ(4)-IREC(37+INDEX))+8*(IREC(25+INDEX)-IRE
         #          C(1+INDEX)))+128
          IRDOT(1)=(1.6667)*((LASTR(3)-IREC(26+INDEX))+8*(IREC(14+INDEX)-LAS
         #          TR(4)))+128
          IRDOT(2)=(1.6667)*((LASTR(4)-IREC(38+INDEX))+8*(IREC(26+INDEX)-IRE
         #          C(2+INDEX)))+128
```

TABLE A.2 (Continued)

PROGRAM COMPA2


```
      DO 60 M=3,18
         IQDOT(M)=(1.6667)*((IREC((M-3)*12+1+INDEX)-IREC((M+2)*12+1+INDE
     #             X))+8*(IREC((M+1)*12+1+INDEX)-IREC((M-1)*12+1+INDEX)))+1
     #             28
         IRDOT(M)=(1.6667)*((IREC((M-3)*12+2+INDEX)-IREC((M+2)*12+2+INDE
     #             X))+8*(IREC((M+1)*12+2+INDEX)-IREC((M-1)*12+2+INDEX)))+1
     #             28
   60 CONTINUE
      CALL PRT2
      ISWTCH2=1
      DO 70 M=19,20
         IQDOT(M)=128
         IRDOT(M)=128
   70 CONTINUE
      CALL PRT2
      IOVALT=IREC(40+INDEX)
      IOVAS=IREC(47+INDEX)
      IOVE1=IREC(64+INDEX)
      IOVE5=IREC(112+INDEX)
      RETURN
C
C     ****************************************************************
C     ACTIVE DATA FOUND AND FOLLOWS ANOTHER ACTIVE SECOND.
C     CALCULATE LAST 2 VALUES OF Q.-R. FOR 19-20 OF PREV-
C     IOUS RECORD.  STORE INTO TYPE3, GO ON AND CALCULATE
C     Q.-R. FOR 1-18 NOT TO BE STORED UNTIL 19-20 ARE
C     CALCULATED.
C     ****************************************************************
C
  110 IQDOT(19)=(1.6667)*((LASTQ(1)-IREC(1+INDEX))+8*(LASTQ(4)-LASTQ(2))
     #          )+128
      IQDOT(20)=(1.6667)*((LASTQ(2)-IREC(13+INDEX))+8*(IREC(1+INDEX)-LAS
     #          TQ(3)))+128
      IRDOT(19)=(1.6667)*((LASTR(1)-IREC(2+INDEX))+8*(LASTR(4)-LASTR(2))
     #          )+128
      IRDOT(20)=(1.6667)*((LASTR(2)-IREC(14+INDEX))+8*(IREC(2+INDEX)-LAS
     #          TR(3)))+128
      ISWTCH2=1
      CALL PRT2
      IQDOT(1)=(1.6667)*((LASTQ(3)-IREC(25+INDEX))+8*(IREC(13+INDEX)-LAS
     #          TQ(4)))+128
      IQDOT(2)=(1.6667)*((LASTQ(4)-IREC(37+INDEX))+8*(IREC(25+INDEX)-IRE
     #          C(1+INDEX)))+128
      IRDOT(1)=(1.6667)*((LASTR(3)-IREC(26+INDEX))+8*(IREC(14+INDEX)-LAS
     #          TR(4)))+128
      IRDOT(2)=(1.6667)*((LASTR(4)-IREC(38+INDEX))+8*(IREC(26+INDEX)-IRE
     #          C(2+INDEX)))+128
      DO 120 M=3,18
         IQDOT(M)=(1.6667)*((IREC((M-3)*12+1+INDEX)-IREC((M+2)*12+1+INDE
     #             X))+8*(IREC((M+1)*12+1+INDEX)-IREC((M-1)*12+1+INDEX)))+1
     #             28
         IRDOT(M)=(1.6667)*((IREC((M-3)*12+2+INDEX)-IREC((M+2)*12+2+INDE
     #            X))+8*(IREC((M+1)*12+2+INDEX)-IREC((M-1)*12+2+INDEX)))+12
     #            8
  120 CONTINUE
      CALL PRT2
      RETURN
```

TABLE A.2 (Continued)

PROGRAM COMPA2

```
SUBROUTINE ACTDATA    74/74    OPT=1 TRACE                FTN [illegible]          10/19/[illegible]  [illegible]       PAGE

                END
```

TABLE A.2 (Continued)

PROGRAM COMPA2


```
      SUBROUTINE PRT1
      DIMENSION IP1(38), IRECO1(87), ITBL1(87), INDD(12),LIST(17)
      COMMON INDEX, IREC(486)
      COMMON/CNTR/ KCUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON /CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON /TCNTR/ KTKNT,LTKNT,NTKNT,KTNTR2
      COMMON /TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON /CM3/ IAC
      COMMON /CM4/ ITH
      COMMON /SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      COMMON /THRES/ ITQ,ITR,ITWY,ITWZ,ITALT,ITAS
      DATA ITBL1/4,-24,52,12*8,4,6*8,12,7*8,4,7*8,4,7*8,4,7*8,4,7*8,4,
     #          7*8,4,7*8,4,7*8,4/
      DATA INDD/48,88,95,96,136,143,144,184,191,192,232,239/
      DATA LIST/1,2,3,4,5,6,7,8,9,10,15,16,23,24,40,47,208/
C*************SWTCH1 DEFINITION*********************************
C    ISWTCH1=0 WHEN IT HAS GONE THROUGH PREVIOUS CALDATA
C    ISWTCH1=1 WHEN IT HAS FOUND THE FIRST GOOD SEC. OF DATA--
C              PRINT TYPE 1 RECORD.
C    ISWTCH1=2 WHEN IT IS TO PROCESS CALDATA THIS
C              ALLOWS FOR INITIALIZATION INTO ARRAY.
C****************************************************************
      IF(ISWTCH1.EQ.1) GO TO 100
      IF(ISWTCH1.EQ.2) INCRST1=24
      INCR=INCRST1-1
C
C  ************** CALIBRATION INFORMATION ****************
C
      DO 50 M=1,17
         INCR=INCR+1
         JINCR=INCR+1
         IF(MOD(JINCR,8).NE.0)GO TO 40
         IRECO1(INCR)=0
         INCR=INCR+1
   40    IRECO1(INCR)=IREC(LIST(M)+INDEX)
   50 CONTINUE
      ISWTCH1=0
      IF(INCR.GE.81)ISWTCH1=2
      INCRST1=INCR
      RETURN
  100 INCR=INCRST1-1
C
C  ************* CREATE TYPE3 RECORD ***********************
C
      ISWTCH3=0
      CALL PRT3
      DO 110 M=INCR,87
         IRECO1(M)=0
  110 CONTINUE
      IRECO1(1)=1
      IRECO1(2)=IAC
      IRECO1(3)=0
C
C  *************** DOCUMENTARY DATA ********************
C
      DO 120 M=1,12
         IRECO1(M+3)=IREC(INDD(M)+INDEX)
```

TABLE A.2 (Continued)

PROGRAM COMPA2


```
  120 CONTINUE
      IREC01(16)=0
C
C  ********************** THRESHOLDS * ******** ****************
C
      IREC01(17)=ITALT
      IREC01(18)=ITAS
      IREC01(19)=ITHY
      IREC01(20)=ITHZ
      IREC01(21)=ITQ
      IREC01(22)=ITR
      IREC01(23)=0
      IJ=1
      IK=1
      DO 130 M=1,38
         IP1(M)=0
  130 CONTINUE
      CALL PACK(IREC01,87,IP1,IJ,IK,IIBL1)
      ISWTCH1=2
      KNTR1=KNTR1+1
C     WRITE(7) (IP1(J),J=1,38)
      RETURN
      END
```

TABLE A.2 (Continued)

PROGRAM COMPA2

```
SUBROUTINE PRT2        74/74   OPT=1 TRACE                  FTN 4.7+476         10/19/79  17.24.29        PAGE     1

          SUBROUTINE PRT2
          DIMENSION IP2(38),IRECO2(281),ITBL2(281)
          COMMON INDEX,IREC(486)
          COMMON /CH4/ ITM
          COMMON /CNTR/ KOUNT,LKCOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
          COMMON /SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
          COMMON /TEMP/ IQDOT(20),IRDOT(20)
          DATA ITBL2/4,16,279*8/
C
C ******** DATA IN ACTIVE PERIODS ******************************
C ********** OUTPUTTED IF ACTIVE CONDITIONS ARE MET
C                 OUTPUT 1 SECOND AFTER CRITEREA FAILS    ********
C
          IF(ISWTCH2.EQ.1)GO TO 20
          IRECO2(1)=2
          IRECO2(2)=ITM
          DO 10 M=3,241
             IRECO2(M)=IREC(M-2+INDEX)
       10 CONTINUE
          RETURN
       20 ISWTCH2=0
          DO 30 M=242,261
             IRECO2(M)=IQDOT(M-241)
       30 CONTINUE
          DO 40 M=262,281
             IRECO2(M)=IRDOT(M-261)
       40 CONTINUE
          IJ=IK=1
          DO 50 M=1,38
             IP2(M)=0
       50 CONTINUE
          CALL PACK(IRECO2,281,IP2,IJ,IK,ITBL2)
C         WRITE(7)(IP2(M),M=1,38)
          KNTR2=KNTR2+1
          RETURN
          END
```

TABLE A.2 (Continued)

PROGRAM COMPA2


```
      SUBROUTINE PRT3
      DIMENSION IP3(38),IRECO3(17),ITBL3(17),INDD(12)
      COMMON INDEX,IREC(486)
      COMMON /CM3/ IAC
      COMMON /CM4/ ITM
      COMMON /CNTR/ KOUNT,LKOUNT,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON /CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON /TCNTR/ KTKNT,LTKNT,NTKNT,KTNTR2
      COMMON /TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON /SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      DATA ITBL3/4,-24,52,12*0,4,60/
      DATA INDD/48,88,95,96,136,143,144,184,191,192,232,239/
C
C ******** END OF FLIGHT (TAIL) *************************
C ********* SWTCH3 DEFINITION **************
C       ISWTCH3=0     WHEN INITIAL ENTRY STORE DOC.DATA
C       ISWTCH3=1     WHEN ITS TO SAVE TIME SEC. AND WRITE
C                     TYPE3 RECORD
C ***************************************************************
C
      IF(ISWTCH3.EQ.1)GO TO 11
      IRECO3(1)=3
      IRECO3(2)=IAC
      IRECO3(3)=0
C
C *************** DOCUMENTARY DATA ***********************
C
      DO 10 M=1,12
         IRECO3(M+3)=IREC(INDD(M)+INDEX)
   10 CONTINUE
      ISWTCH3=1
      RETURN
   11 IRECO3(16)=0
      IRECO3(17)=ITM
      IEND=1
      ISWTCH3=0
      IF(KOUNT.EQ.0)GO TO 30
      KNTR3=KNTR3+1
      IJ=IK=1
      DO 20 M=1,38
         IP3(M)=0
   20 CONTINUE
      CALL PACK(IRECO3,17,IP3,IJ,IK,ITBL3)
C     WRITE(7)(IP3(J),J=1,38)
C
C *****************************************************************
C FINAL INDIVIDUAL FLIGHT STATISTICS
C *****************************************************************
C
      WRITE(6,1010)KNTR1
 1010 FORMAT(8H0FLIGHT#,I3,11H STATISTICS)
      WRITE(6,1020)KNTR2
 1020 FORMAT(41H 1.  THE NUMBER OF TYPE2 RECORDS PRINTED#,I7)
      WRITE(6,1060)IGRDCNT
 1060 FORMAT(28H 2.  SECONDS OF GROUND DATA#,I7)
      WRITE(6,1090)KOUNT
 1090 FORMAT(28H 3.  SECONDS OF ACTIVE DATA#,I7)
```

TABLE A.2 (Concluded)

PROGRAM COMPA2


```
      WRITE(6,1110)ITM
 1110 FORMAT(29H 4.  TOTAL NUMBER OF SECONDS:,I7)
   30 KTNTR2=KNTR2+KTNTR2
      ITERCNT=IERRCNT+ITERCNT
      ITSTCNT=ISHTCNT+ITSTCNT
      ITGDCNT=IGRDCNT+ITGDCNT
      ITKNT=IKOUNT+ITKNT
      KTKNT=KOUNT+KTKNT
      LTKNT=LKOUNT+LTKNT
      NTKNT=NTKNT+ITM
      KNTR2=IERRCNT=ISHTCNT=IGRDCNT=0
      IKOUNT=KOUNT=LKOUNT=0
      NKOUNT=NKOUNT-ITM
      RETURN
      END
```

TABLE A.3

PROGRAM COMPB1


```
      PROGRAM COMPB1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE7)
C ***** RON JANNING UDRI 547 93579-00 *****
C ************************************************************
C THIS ROUTINE IS THE SECOND COMPRESSION METHODOLOGY
C DESIGNED FOR THE A10 AIRCRAFT.  USING A FLOATING MEAN
C AND WINDOW VALUES AS DETERMINANTS TO WHETHER A
C PARAMETER IS ACTIVE DATA.
C ************************************************************
C
      DIMENSION NTBLK(324),ITBL(486),LIST(16)
      DIMENSION LASTVAL(12),LISTA(16),LISTB(16)
      COMMON INDEX,IREC(486)
      COMMON/C7/IREC02(96),IDIR(16),IDVALUE(16)
      COMMON/CH3/IAC,MRECNT,INCRST1
      COMMON/SWTCH/ISWTCH1,ISWTCH2,ISWTCH3,IEND
      COMMON/C8/IBOUND(32),LFRAME(32),LSAMPLE(32)
      COMMON/CH4/ITM
      COMMON/CNTR/KOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON/CNTR2/IKOUNT,IGRDCNT,IERRCNT,ISWTCNT
      COMMON/THRES/IWINDOW(16)
      COMMON/TCNTR/KTKNT,NTKNT,KTNTR2
      COMMON/TCNTR2/ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      DATA ITBL/486*0/
      DATA LIST/10,15,13,14,12,8,9,7,6,3,5,11,98,1,16,17/
      DATA LISTA/15,15,15,15,15,13,13,13,15,15,15,15,1,1,1,1/
      DATA LISTB/16,16,16,16,16,16,16,16,16,16,16,16,0,0,0,0/
      IERRCNT=ISWTCNT=IGRDCNT=ITERCNT=ITSTCNT=ITGDCNT=0
      ISWTCH2=ITM=ISWTCH4=INFLGHT=ISWTCH5=IEND=0
      KOUNT=MPEAK=NKOUNT=IKOUNT=NTKNT=KTKNT=0
      KNTR1=KNTR2=KNTR3=KTNTR2=IWINDOW(15)=IWINDOW(16)=0
      MRECNT=ISWTCH3=1
      DO 3 M=1,16
         IDIR(M)=1
    3 CONTINUE
      ISWTCH1=INCRST1=2
      READ(5,1)IAC,NGR,(IWINDOW(N),N=1,14)
    1 FORMAT(A4,I1,14I3)
      WRITE(6,5)IAC
    5 FORMAT(12H1AIRCRAFT:   ,A4)
      IF(NGR.EQ.1)WRITE(6,6)
      IF(NGR.EQ.0)WRITE(6,7)
    6 FORMAT(23H PROCESSING GROUND DATA)
    7 FORMAT(27H NOT PROCESSING GROUND DATA)
      WRITE(6,10)(IWINDOW(N),N=1,8)
   10 FORMAT(52H WINDOW VALUES: NY  NZ  P   Q   R  P-DOT Q-DOT R-DOT,
     1/15X,I3,1X,I3,1X,I3,1X,I3,1X,I3,2X,I3,3X,I3,3X,I3)
      WRITE(6,15)(IWINDOW(N),N=9,14)
   15 FORMAT(41H               FCTR PLA DSB SIGMA ALT  AS,
     1/16X,I3,1X,I3,1X,I3,2X,I3,2X,I3,1X,I3)
C
C READING IN 10 SECONDS
C
20    BUFFER IN(1,1)(NTBLK(1),NTBLK(324))
      IF(UNIT(1))25,900,22
22    NKOUNT=NKOUNT+10
      GO TO 20
25    NKOUNT=NKOUNT+10
```

TABLE A.3 (Continued)

PROGRAM COMPB1


```
      IM=JM=1
C
C PROCESSING THE 10 SECONDS 2 AT A TIME
C
      DO 100 K=1,5
        CALL UNPACK(IREC,486,NTBLK,IM,JM,ITBL)
        IF(ISWTCH4.EQ.1)GO TO 93
C
C PARITY ERROR CHECK
C
        IF(OR(SHIFT(IREC(483),8),IREC(484)).NE.0)GO TO 91
C
C  SHORT RECORD CHECK
C
        IF(AND(IREC(480),15).NE.9)GO TO 92
        INDEX=0
        DO 90 L=1,2
        IF(L.EQ.2)INDEX=240
C
C BUILT IN CYCLE TEST
C
          JBIT=IREC(INDEX+114)/128
          JCAL=AND((IREC(INDEX+114)/64),1)
          JPRE=AND((IREC(INDEX+114)/32),1)
          IF(JBIT.EQ.1.AND.INFLGHT.NE.32.AND.IEND.NE.1)CALL PRT3
          IF(JCAL.EQ.1.OR.JBIT.EQ.1.OR.JPRE.EQ.1)IKOUNT=IKOUNT+1
          IF(JCAL.EQ.1.OR.JPRE.EQ.1)GO TO 87
          IF(JBIT.EQ.1)GO TO 86
          INFLGHT=AND(IREC(16+INDEX),32)
          ITM=NKOUNT-10+2*(K-1)+L
          IEND=0
C
C FLIGHT CHECK
C
          IF(INFLGHT.EQ.0)IGRDCNT=IGRDCNT+1
          IF(NGR.EQ.1)GO TO 26
          IF(INFLGHT.EQ.0)GO TO 87
C
C DONE ONCE AFTER BIT CYCLE HAS BEEN SHUT DOWN.
C
   26     IF(ISWTCH1.NE.0)GO TO 30
          ISWTCH1=2
          DO 28 M=1,16
            N2=LIST(M)
            ISTO=LIST(M)+INDEX
            IF(M.LT.6.OR.M.GT.8)GO TO 27
            IREC(ISTO)=128
   27       IOVALUE(M)=IREC(ISTO)
            IBOUND(M)=IBOUND(M+16)=IOVALUE(M)
            LFRAME(M)=LFRAME(M+16)=ITM
            LSAMPLE(M)=LSAMPLE(M+16)=N2
   28       CONTINUE
            ITEMPTM=ITM
C
C  CALCULATION OF LAST 2 PITCH, ROLL, AND YAW VALUES
C  AFTER PREVIOUS CALCULATIONS
C  ISWICH5 IS A SWITCH TO DETERMINE
```

TABLE A.3 (Continued)

PROGRAM COMPB1


```
      C  IF THERE IS A PREVIOUS CALCULATION.
      C   ISWTCH5=0 WHEN THERE IS NO PREVIOUS LASTVAL
      C            VALUES HAVE BEEN STORED.
      C   ISWTCH5=1 WHEN THERE ARE PREVIOUS VALUES FOR
      C            CALCULATIONS OF PITCH, YAW AND ROLL
      C            SLOPE VALUES OF THE 14TH AND 15TH OF
      C            THE PREVIOUS SECOND AND ALSO THE 1ST
      C            AND 2NC VALUES OF THE PRESENT SECOND.
      C
      C
   30       IF(ISWTCH5.EQ.0)GO TO 56
            DO 35 N=1,3
               IREC(N+214+INDEX)=1.25*((LASTVAL(N*4-3)-IREC(N+11+INDEX))
     1                   +8*(LASTVAL(N*4)-LASTVAL(N*4-2)))+128
   35       CONTINUE
            DO 40 N=1,3
               IREC(N+230+INDEX)=1.25*((LASTVAL(N*4-2)-IREC(N+27+INDEX))
     1                   +8*(IREC(N+11+INDEX)-LASTVAL(N*4-1)))+128
   40       CONTINUE
      C
      C  CHECK MAX/MIN VALUES OF LAST TWO PITCH, ROLL, AND YAW
      C  VALUES THEN CHECK TO SEE IF IT IS A WINDOW
      C  CROSSING
      C
            DO 50 N=6,8
               ITEMP=LIST(N)+208+INDEX
               JTEMP=ITEMP+16
               DO 50 M=ITEMP,JTEMP,16
                  IF(IREC(M).GE.IBOUND(N))GO TO 42
                  IBOUND(N)=IREC(M)
                  LFRAME(N)=ITM-1
                  LSAMPLE(N)=M+5-INDEX
   42             IF(IREC(M).LE.IBOUND(N+16))GO TO 44
                  IBCUND(N+16)=IREC(M)
                  LFRAME(N+16)=ITM-1
                  LSAMPLE(N+16)=M+5-INDEX
   44             N1=ITM-1
                  N2=M+5-INDEX
                  IF(IREC(M).LE.IOVALUE(N)+IWINDOW(N))GO TO 47
                    IF(KOUNT.NE.0)GO TO 46
                    ISWTCH1=1
                    CALL PRT1
                    DO 45 M4=1,16
                    CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1)
   45               CONTINUE
   46               CALL PRT2(N,N1,N2,IREC(M),MRECNT,1)
   47             IF(IREC(M).GE.IOVALUE(N)-IWINDOW(N))GO TO 50
                    IF(KOUNT.NE.0)GO TO 49
                    ISWTCH1=1
                    CALL PRT1
                    DO 48 M4=1,16
                    CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1)
   48               CONTINUE
   49               CALL PRT2(N,N1,N2,IREC(M),MRECNT,0)
   50       CONTINUE
      C
      C  CALCULATION OF PITCH, RCLL, AND YAW 1-13 SAMPLES
```

TABLE A.3 (Continued)

PROGRAM COMPB1


```
      C
            DO 52 N=1,3
               IREC(N+6+INDEX)=1.25*((LASTVAL(N*4-1)-IREC(N+43+INDEX))
           1         +8*(IREC(N+27+INDEX)-LASTVAL(N*4)))+128
         52 CONTINUE
            DO 54 N=1,3
               IREC(N+22+INDEX)=1.25*((LASTVAL(N*4)-IREC(N+59+INDEX))
           1         +8*(IREC(N+43+INDEX)-IREC(N+11+INDEX)))+128
         54 CONTINUE
         56 DO 58 N=6,8
               ITEMP=LIST(N)+32+INDEX
               JTEMP=ITEMP+160
               DO 58 M=ITEMP,JTEMP,16
                  IREC(M)=1.25*((IREC(M-27)-IREC(M+37))+8*
           1               (IREC(M+21)-IREC(M-11)))+128
         58 CONTINUE
         DO 70 M1=1,16
         N1=LISTA(M1)
      C
      C CHECK MAX/MIN VALUES FOR ANY POSSIBLE
      C CHANGE AND CHECK FOR WINDOW CROSSING
      C
         DO 69 M2=1,N1
            IF(M1.LT.6.OR.M1.GT.8)GO TO 61
            IF(ISWTCH5.EQ.0.AND.M2.LT.3)GO TO 69
         61 N2=LIST(M1)+((M2-1)*LISTB(M1))
            ISTO=N2+INDEX
            IF(M1.GT.5.AND.M1.LT.9)N2=N2+5
            IF(IREC(ISTO).GE.IBOUND(M1))GO TO 62
            IBOUND(M1)=IREC(ISTO)
            LFRAME(M1)=ITM
            LSAMPLE(M1)=N2
         62 IF(IREC(ISTO).LE.IBOUND(M1+16))GO TO 63
            IBOUND(M1+16)=IREC(ISTO)
            LFRAME(M1+16)=ITM
            LSAMPLE(M1+16)=N2
         63 IF(IREC(ISTO).LE.IOVALUE(M1)+IWINDOW(M1))GO TO 66
            IF(KOUNT.NE.0)GO TO 65
            ISWTCH1=1
            CALL PRT1
            DO 64 M4=1,16
            CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1)
         64 CONTINUE
         65 CALL PRT2(M1,ITM,N2,IREC(ISTO),MRECNT,1)
         66 IF(IREC(ISTO).GE.IOVALUE(M1)-IWINDOW(M1))GO TO 69
            IF(KOUNT.NE.0)GO TO 68
            ISWTCH1=1
            CALL PRT1
            DO 67 M4=1,16
            CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1)
         67 CONTINUE
         68 CALL PRT2(M1,ITM,N2,IREC(ISTO),MRECNT,0)
            IF(IREC(ISTO).LT.IOVALUE(M1)-IWINDOW(M1))
           1   CALL PRT2(M1,ITM,N2,IREC(ISTO),MRECNT,0)
         69 CONTINUE
         70 CONTINUE
         N=0
```

TABLE A.3 (Continued)

PROGRAM COMPB1

```
PROGRAM COMPB1      74/74   OPT=1 TRACE                    FTN 4.7+476        11/13/79  17.37.24       PAGE    5

                    DO 75 M=187,235,16
                       N=N+1
                       DO 75 M1=1,3
                          LASTVAL(N+4*(M1-1))=IREC(M+M1+INDEX)
               75   CONTINUE
                    ISWTCH5=1
                     GO TO 90
          86         IF(INFLGHT.NE.32)CALL PRT1
          87         ISWTCH5=0
          90         CONTINUE
                   GO TO 100
          91       IERRCNT=IERRCNT+2
                   ISWTCH5=0
                   GO TO 100
          92       ISHTCNT=ISHTCNT+2
                   ISWTCH5=0
                   GO TO 100
          93       ISWTCH4=1
          100      CONTINUE
                 GO TO 20
          C
          C ANY INCOMPLETE TYPE2 DATA? WRITE LAST TYPE2 RECORD.
          C
          900    ISWTCH3=1
                 CALL PRT3
                 WRITE(6,1010)
          1010   FORMAT(32H1PRINTOUT FOR COMPB1 STATISTICS*     )
                 WRITE(6,1050)KNTR1
          1050   FORMAT(41H01.  THE NUMBER OF TYPE1 RECORDS WRITTEN*      ,I7)
                 WRITE(6,1060)KTNTR2
          1060   FORMAT(41H02.  THE NUMBER OF TYPE2 RECORDS WRITTEN*      ,I7)
                 WRITE(6,1070)KNTR3
          1070   FORMAT(41H03.  THE NUMBER OF TYPE3 RECORDS WRITTEN*      ,I7)
                 WRITE(6,1074)ITGDCNT
           1074  FORMAT(28H04.  SECONDS OF GROUND DATA*,I7)
                 WRITE(6,1076)KTKNT
           1076  FORMAT(26H05.  TOTAL ACTIVE SAMPLES*,I7)
                 WRITE (6,1080) MPEAK
          1080   FORMAT(35H06.  THE NUMBER OF PEAKS STORED*     ,I7)
                 WRITE(6,1100)NTKNT
          1100   FORMAT(35H07.  THE TOTAL NUMBER OF SECONDS*       ,I8)
                 STOP
                 END
```

TABLE A.3 (Continued)

PROGRAM COMPB1


```
      SUBROUTINE PRT1
      DIMENSION IP1(16),IRECO1(72),ITBL1(72),LIST(11)
      COMMON INDEX,IREC(486)
      COMMON/CNTR/ KOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON/CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON/TCNTR/ KTKNT,NTKNT,KTNTR2
      COMMON/TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON/CM3/ IAC,MRECNT,INCRST1
      COMMON/CM4/ ITM
      COMMON/C7/ IRECO2(96),IDIR(16),IOVALUE(16)
      COMMON/SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      COMMON/THRES/IWINDOW(16)
      DATA ITBL1/4,-24,52,12*8,4,7*8,4,7*8,4,7*8,4,7*8,4,7*8,4,7*8,4,
     1  7*8,4/
      DATA LIST/10,15,13,14,12,5,11,19,1,98,6/
C
C    ****REPEAT BEFORE EACH FLIGHT...IE.FIRST RECORD****
C      ********SWITCH1 DEFINITION********
C        ISWTCH1=0 WHEN IT HAS GONE THROUGH PREVIOUS CAL.
C        ISWTCH1=1 WHEN IT HAS FOUND THE FIRST GOOD SEC.
C                  OF DATA---PRINT TYPE 1 RECORD.
C        ISWTCH1=2 WHEN IT IS TO PROCESS CAL DATA THIS
C                  ALLOWS FOR INITIALIZATION INTO ARRAY.
C      ***********************************
      IF(ISWTCH1.EQ.1)GO TO 135
      IF(ISWTCH1.EQ.2) INCRST1=33
      INCR=INCRST1-1
C
C   ****************CALIBRATED DATA *******************
C
      DO 130 M1=1,11
         INCR=INCR+1
         IF(MOD(INCR,8).NE.0)GO TO 20
         IRECO1(INCR)=0
         INCR=INCR+1
   20    IRECO1(INCR)=IREC(LIST(M1)+INDEX)
  130    CONTINUE
      ISWTCH1=0
      IF(INCR.GE.69)ISWTCH1=2
      INCRST1=INCR
      RETURN
  135 INCR=INCRST1-1
      ISWTCH3=0
      CALL PRT3
      ISWTCH3=1
      DO 137 M=INCR,66
          IRECO1(M)=0
  137 CONTINUE
      IRECO1(1)=1
      IRECO1(2)=IAC
      IRECO1(3)=0
C
C   *****DOCUMENTARY DATA*****
C
      INCR=4
      DO 140 M=63,127,16
         DO 140 N=1,3
```

TABLE A.3 (Continued)

PROGRAM COMPB1


```
                  IF(M+N.EQ.98.OR.M+N.GT.128)GO TO 140
                  IREC01(INCR)=IREC(M+N+INDEX)
                  INCR=INCR+1
         140 CONTINUE
             IREC01(16)=0
      C
      C    ****WINDOW VALUES****
      C
             IREC01(17)=IWINDOW(1)
             IREC01(18)=IWINDOW(2)
             IREC01(19)=IWINDOW(3)
             IREC01(20)=IWINDOW(4)
             IREC01(21)=IWINDOW(5)
             IREC01(22)=IWINDOW(6)
             IREC01(23)=IWINDOW(7)
             IREC01(24)=0
             IREC01(25)=IWINDOW(8)
             IREC01(26)=IWINDOW(11)
             IREC01(27)=IWINDOW(12)
             IREC01(28)=IWINDOW(10)
             IREC01(29)=IWINDOW(14)
             IREC01(30)=IWINDOW(13)
             IREC01(31)=IWINDOW(9)
             IREC01(32)=0
             IK=IJ=1
             DO 155 M=1,16
                IP1(M)=0
         155 CONTINUE
             CALL PACK(IREC01,72,IP1,IJ,IK,ITBL1)
             ISWTCH1=2
             KNTR1=KNTR1+1
      C      WRITE(7)(IP1(J),J=1,16)
             RETURN
             END
```

TABLE A.3 (Continued)

PROGRAM COMPB1


```
      SUBROUTINE PRT2(IDENT,NFRAME,NTIME,JVALUE,KRECPN,IW)
      DIMENSION IPACK(96),IP2(16)
      COMMON/C7/IRECO2(96),IDIR(16),IOVALUE(16)
      COMMON/C8/IBOUND(32),LFRAME(32),LSAMPLE(32)
      COMMON/CNTR/KOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3
      DATA IPACK/4,3*12,23*(8,16,8,8)/
      IP=IB=1
      IF(IW.EQ.-1)GO TO 200
      IF(IW.EQ.IDIR(IDENT))GO TO 200
C
C STORAGE OF PEAKS IN THE TYPE2 RECORDS:
C    IDENTIFICATION, SECONDS, PARAMETER POSITION
C    AND PEAK VALUE.
C
      IRECO2(4*(KRECPN)+1)=IDENT
      IRECO2(4*(KRECPN)+2)=LFRAME(IDIR(IDENT)*16+IDENT)
      IRECO2(4*(KRECPN)+3)=LSAMPLE(ICIR(IDENT)*16+IDENT)
      IRECO2(4*(KRECPN)+4)=IBOUND(IDIR(IDENT)*16+IDENT)
      IDIR(IDENT)=IW
      KOUNT=KOUNT+1
      MPEAK=MPEAK+1
      IF(KRECPN.LT.23)GO TO 300
      IRECO2(1)=2
      IRECO2(2)=IRECO2(3)=IRECO2(4)=0
      DO 50 M=1,16
         IP2(M)=0
   50 CONTINUE
      CALL PACK(IRECO2,96,IP2,IP,IB,IPACK)
C     WRITE(7)(IP2(I),I=1,16)
      KNTR2=KNTR2+1
      KRECPN=0
  300 KRECPN=KRECPN+1
C
C STORAGE OF WINDOW CROSSINGS:
C    IDENTIFICATION, SECONDS, PARAMETER POSITION
C    AND WINDOW CROSSING VALUES.
C
  200 IRECO2(4*(KRECPN)+1)=IDENT
      IRECO2(4*(KRECPN)+2)=NFRAME
      IRECO2(4*(KRECPN)+3)=NTIME
      IRECO2(4*(KRECPN)+4)=JVALUE
      LFRAME(IDENT)=LFRAME(IDENT+16)=NFRAME
      LSAMPLE(IDENT)=LSAMPLE(IDENT+16)=NTIME
      IBOUND(IDENT)=IBOUND(IDENT+16)=JVALUE
      IOVALUE(IDENT)=JVALUE
      KOUNT=KOUNT+1
      IF(KRECPN.LT.23)GO TO 100
      IP=IB=1
      IRECO2(1)=2
      IRECO2(2)=IRECO2(3)=IRECO2(4)=0
      DO 150 M=1,16
         IP2(M)=0
  150 CONTINUE
      CALL PACK(IRECO2,96,IP2,IP,IB,IPACK)
C     WRITE(7)(IP2(I),I=1,16)
      KNTR2=KNTR2+1
      KRECPN=0
```

TABLE A.3 (Continued)

PROGRAM COMPB1


```
      100 KRECPN=KRECPN+1
          RETURN
          END
```

TABLE A.3 (Continued)

PROGRAM COMPB1


```
      SUBROUTINE PRT3
      DIMENSION IP3(16),IRECO3(17),ITBL3(17)
      COMMON INDEX,IREC(486)
      COMMON/C7/IRECO2(96),IDIR,IOVALUE(16)
      COMMON/CM3/ IAC,MRECNT,INCRST1
      COMMON/CM4/ ITM
      COMMON/CNTR/ KCOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON/CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON/TCNTR/ KTKNT,NTKNT,KTNTR2
      COMMON/TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON/SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      DATA ITBL3/4,-24,52,12*8,4,60/
C
C    ****END OF FLIGHT (TAIL)****
C     *****SWITCH3 DEFINTION*******
C      ISWTCH3=0 WHEN INITIAL ENTRY STOR DOC.DATA.
C      ISWTCH3=1 WHEN ITS TO SAVE TIME SEC. AND WRITE TYPE 3 REC.
C     ******************************
      IF (ISWTCH3.EQ.1)GO TO 11
      IRECO3(1)=3
      IRECO3(2)=IAC
      IRECO3(3)=0
C
C    ****DOCUMENTARY DATA****
      INCR=4
      DO 10 M=63,127,16
         DO 10 N=1,3
            IF(M+N.EQ.98.OR.M+N.GT.128)GO TO 10
            IRECO3(INCR)=IREC(M+N+INDEX)
            INCR=INCR+1
   10 CONTINUE
      ISWTCH3=1
      RETURN
   11 IRECO3(16)=0
      IRECO3(17)=ITM
      IEND=1
      ISWTCH3=0
      IF(KOUNT.EQ.0)GO TO 30
C
C CLEAR OUT REMAINING TYPE2 RECORDS
C BEFORE WRITING OUT THE TYPE3 RECORD.
C
      IF(MRECNT.EQ.1)GO TO 18
      IF(MRECNT.EQ.23)GO TO 15
      DO 12 K7=MRECNT,23
         IRECO2(4*(K7)+1)=IRECO2(4*(K7)+2)=0
         IRECO2(4*(K7)+3)=IRECO2(4*(K7)+4)=0
   12 CONTINUE
      MRECNT=23
   15 CALL PRT2(1,0,0,0,MRECNT,-1)
      KOUNT=KOUNT-1
   18 KNTR3=KNTR3+1
      IK=IJ=1
      DO 20 M=1,16
         IP3(M)=0
   20 CONTINUE
      CALL PACK(IRECO3,17,IP3,IJ,IK,ITBL3)
```

TABLE A.3 (Concluded)

PROGRAM COMPB1


```
C     WRITE(7)(IP3(J),J=1,16)
      WRITE(6,1010)KNTR1
 1010 FORMAT(8HOFLIGHT#,I3,11H STATISTICS)
      WRITE(6,1020)KNTR2
 1020 FORMAT(41H 1.  THE NUMBER OF TYPE2 RECORDS WRITTEN#,I7)
      WRITE(6,1060)IGRDCNT
 1060 FORMAT(28H 2.  SECONDS OF GROUND DATA#,I7)
      WRITE(6,1090)KOUNT
 1090 FORMAT(28H 3.  SAMPLES OF ACTIVE DATA#,I7)
      WRITE(6,1110)ITM
 1110 FORMAT(29H 4.  TOTAL NUMBER OF SECONDS#,I7)
   30 NKOUNT=NKOUNT-ITM
      KTNTR2=KNTR2+KTNTR2
      ITERCNT=IERRCNT+ITERCNT
      ITSTCNT=ISHTCNT+ITSTCNT
      ITGDCNT=IGRDCNT+ITGDCNT
      ITKNT=IKOUNT+ITKNT
      KTKNT=KOUNT+KTKNT
      NTKNT=ITM+NTKNT
      KNTR2=IERRCNT=ISHTCNT=IGRDNT=0
      IKOUNT=KOUNT=LKOUNT=0
      RETURN
      END
```

TABLE A.4

PROGRAM COMPB2


```
      PROGRAM COMPB2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE7)
C ***** RON JANNING UDRI 547 93579-00 *****
C**********************************************************************
C     METHOD B ON C141 & C130 MT!S
C**********************************************************************
      DIMENSION NTBLK(324),ITBL(486),LIST(21),LISTA(21),LISTB(21)
      DIMENSION LASTVAL(8),ICOT(40)
      COMMON INDEX,IREC(486)
      COMMON/C7/IREC02(96),IDIR(21),IOVALUE(21)
      COMMON/CH3/IAC,NRECNT
      COMMON/SWTCH/ISWTCH1,ISWTCH2,ISWTCH3,IEND
      COMMON/C8/IBOUND(42),LFRAME(42),LSAMPLE(42)
      COMMON/CH4/ITM
      COMMON/CNTR/KOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3
      COMMON/CNTR2/IKOUNT,IGROCNT,IERRCNT,ISHTCNT
      COMMON/THRES/IWINDOW(21)
      COMMON/TCNTR/KTKNT,NTKNT,KTNTR2
      COMMON/TCNTR2/ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      DATA ITBL/486*0/
      DATA LIST/4,3,1,2,1,2,5,6,9,10,7,8,15,23,24,16,40,47,208,64,112/
      DATA LISTA/30,30,20,20,18,18,20,20,20,20,10,10,10,5,5,2,1,1,1,1,1/
      DATA LISTB/8,8,12,12,0,0,12,12,12,12,24,24,24,48,48,144,0,0,0,0,0/
      IERRCNT=ISHTCNT=IGROCNT=ITERCNT=ITSTCNT=ITGDCNT=0
      IWINDOW(20)=IWINDOW(21)=IEND=0
      ISWTCH2=ITM=ISWTCH4=ISWTCH5=KOUNT=MPEAK=NKOUNT=IKOUNT=0
      INFLGHT=KNTR1=KNTR2=KNTR3=KTNTR2=NTKNT=KTKNT=0
      NRECNT=ISWTCH3=1
      DO 3 M=1,21
         IDIR(M)=1
    3 CONTINUE
      ISWTCH1=2
      READ(5,1)IAC,NGR,(IWINDOW(N),N=1,19)
1     FORMAT(A4,I1,19I3)
      WRITE(6,5)IAC
    5 FORMAT(10H1AIRCRAFT!,A4)
      IF(NGR.EQ.1)WRITE(6,6)
      IF(NGR.EQ.0)WRITE(6,7)
    6 FORMAT(23H PROCESSING GROUND DATA)
    7 FORMAT(27H NOT PROCESSING GROUND DATA)
      WRITE(6,8)(IWINDOW(N),N=1,11)
    8 FORMAT(61H WINDOW VALUES! NY! NZ  Q    R Q-DOT R-DOT   S2   S3   S4   S
     #5   S6,/15X,4(I3,1X),I3,3X,I3,2X,5(I3,1X))
      WRITE(6,9)(IWINDOW(N),N=12,19)
    9 FORMAT(47H                     DLR DLE DLF DLNG  VG DLP  PS  PT,
     #/15X,I3,1X,I3,1X,I3,2X,I3,1X,I3,1X,I3,1X,I3,1X,I3)
C
C READING IN 10 SECONDS
C
20    BUFFER IN(1,1)(NTBLK(1),NTBLK(324))
      IF(UNIT(1))25,900,22
22    NKOUNT=NKOUNT+10
      GO TO 20
25    NKOUNT=NKOUNT+10
      IM=JM=1
C
C PROCESSING THE 10 SECONDS 2 AT A TIME
C
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
      DO 100 K=1,5
        CALL UNPACK(IREC,486,NTBLK,IM,JM,ITBL)
        IF(ISWTCH4.EQ.0)GO TO 93
C
C PARITY ERROR CHECK
C
      IF(OR(SHIFT(IREC(483),8),IREC(484)).NE.0) GO TO 91
C
C  SHORT RECORD CHECK
C
      IF(AND(IREC(480),15).NE.9) GO TO 92
        INDEX=0
        DO 90 L=1,2
          IF(L.EQ.2) INDEX=240
C
C BUILT IN CYCLE TEST..
C
          JBIT=IREC(INDEX+232)/128
          JCAL=AND((IREC(INDEX+232)/64),1)
          JPRE=AND((IREC(INDEX+232)/32),1)
          IF(JBIT.EQ.1.AND.INFLGHT.NE.64.AND IEND.NE.1)CALL PRT3
          IF(JCAL.EQ.1.OR.JBIT.EQ.1.OR.JPRE.EQ.1)IKOUNT=IKOUNT+1
          IF(JCAL.EQ.1.OR.JPRE.EQ.1)GO TO 87
          IF(JBIT.EQ.1)GO TO 86
          ITM=NKOUNT-10+2*(K-1)+L
          IEND=0
          INFLGHT=AND(IREC(64+INDEX),64)
C
C  FLIGHT CHECK
C
          IF(INFLGHT.EQ.0)IGRDCNT=IGRDCNT+1
          IF(NGR.EQ.1)GO TO 26
          IF(INFLGHT.EQ.0)GO TO 87
C
C DONE ONCE AFTER BIT CYCLE HAS BEEN SHUT DOWN.
C
   26     IF(ISWTCH1.NE.0)GO TO 30
          ISWTCH1=2
          DO 29 M=1,21
            IF(M.LT.5.OR.M.GT.6)GO TO 27
            N2=LIST(M-2)
            ISTO=N2+INDEX
            IOVALUE(M)=128
            GO TO 28
   27       N2=LIST(M)
            ISTO=N2+INDEX
            IOVALUE(M)=IREC(ISTO)
   28       IBOUND(M)=IBOUND(M+21)=IOVALUE(M)
            LFRAME(M)=LFRAME(M+21)=ITM
            LSAMPLE(M)=LSAMPLE(M+21)=ISTO
            ITEMPT(=ITM
   29       CONTINUE
C
C ANANLYSIS OF ALL 21 PARAMETERS OF THE SECOND.
C  1. CALCULATE THE PITCH AND YAW SLOPE VALUES OF 19-20
C  2. ANANLYZE THE MAX/MIN CHECKS AND THE WINDOW CROSINGS OF 19-20
C  3. CALCULATE THE PITCH AND YAW SLOPE VALUES OF SAMPLES 1-18 OF THE
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
C      PRESENT SECOND.
C   4. ANALYZE THE MAX/MIN CHECKS AND ALL WINDOW CROSSING OF ALL
C      PARAMETERS.
C
  30         IF(ISWTCH5.EQ.0)GO TO 45
C
C DEFINITION OF ISWTCH5
C  ISWTCH5=0 WHEN NO PREVIOUS LASTVAL VALUES HAVE BEEN STORED.
C  ISWTCH5=1 WHEN THERE ARE PREVIOUS VALUES FOR THE CALCULATION
C            OF THE PITCH AND YAW SLOPE VALUES OF THE 19TH AND 20TH
C            SAMPLE VALUES OF THE PREVIOUS SECOND AND 1ST AND 2ND
C            VALUES OF THE PRESENT SECOND.
C
             DO 32 N=1,2
                IDOT(N*20-1)=1.6667*((LASTVAL(N*4-3)-IREC(N+INDEX))
    1                +8*(LASTVAL(N*4)-LASTVAL(N*4-2)))+128
  32         CONTINUE
             DO 33 N=1,2
                IDOT(N*20)=1.6667*((LASTVAL(N*4-2)-IREC(N+12+INDEX))
    1                +8*(IREC(N+INDEX)-LASTVAL(N*4-1)))+128
  33         CONTINUE
             N2=ITM-1
             DO 43 N=18,38,20
                N1=4
                DO 43 M=1,2
                   N1=N1+1
                   IF(IDOT(N+M).GE.IBOUND(N1))GO TO 34
                   IBOUND(N1)=IDOT(N+M)
                   LFRAME(N1)=N2
                   LSAMPLE(N1)=LIST(N1-2)+(M-1)*12+216
  34               IF(IDOT(N+M).LE.IBOUND(N1+21))GO TO 35
                   IBOUND(N1+21)=IDOT(N+M)
                   LFRAME(N1+21)=N2
                   LSAMPLE(N1+21)=LIST(N1-2)+(M-1)*12+216
  35               N3=LIST(N1-2)+(M-1)*12+216
                   IF(IDOT(N+M).LE.IOVALUE(N1)+IWINDOW(N1))GO TO 38
                   IF(KOUNT.NE.0)GO TO 37
                   ISWTCH1=1
                   CALL PRT1
                   DO 36 M4=1,21
                   CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1
    1                       )
  36               CONTINUE
  37               CALL PRT2(N1,N2,N3,IDOT(N+M),MRECNT,1)
  38               IF(IDOT(N+M).GE.IOVALUE(N1)-IWINDOW(N1))GO TO 43
                   IF(KOUNT.NE.0)GO TO 41
                   ISWTCH1=1
                   CALL PRT1
                   DO 39 M4=1,21
                   CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1
    1                      )
  39               CONTINUE
  41               CALL PRT2(N1,N2,N3,IDOT(N+M),MRECNT,0)
  43            CONTINUE
                DO 44 N=1,2
                   IDOT(N*20-19)=1.6667 ((LASTVAL(N*4-1)-IREC(N+24+
    1                 INDEX))+8*(IREC(N+12+INDEX)-LASTVAL(N 4)))+128
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
       44       CONTINUE
                DO 45 N=1,2
                   IDOT(N*20-18)=1.6667*((LASTVAL(N*4)-IREC(N+36+INDEX
          1          ))+8*(IREC(N+24+INDEX)-IREC(N+INDEX)))+128
       45       CONTINUE
       46       DO 48 N=3,18
                   N2=2
                   DO 48 M=1,2
                           N2=N2+1
                     IDOT(N+20*(M-1))=1.6667*((IREC(LIST(N2)+(N-3)*12
          1                )-IREC(LIST(N2)+(N+1)*12))+8*(IREC(LIST(N2)+N
          1                *12)-IREC(LIST(N2)+(N-2)*12)))+128
       48       CONTINUE
       49    DO 69 M1=1,21
                N1=LISTA(M1)
                IF(M1.GT.4.AND.M1.LT.7)GO TO 59
                DO 58 M2=1,N1
                   N2=LIST(M1)+((M2-1)*LISTB(M1))
                   ISTO=N2+INDEX
                   IF(IREC(ISTO).GE.IBOUND(M1)) GO TO 50
                   IBOUND(M1)=IREC(ISTO)
                   LFRAME(M1)=ITM
                   LSAMPLE(M1)=N2
       50          IF(IREC(ISTO).LE.IBOUND(M1+21))GO TO 51
                   IBOUND(M1+21)=IREC(ISTO)
                   LFRAME(M1+21)=ITM
                   LSAMPLE(M1+21)=N2
       51          IF(IREC(ISTO).LE.IOVALUE(M1)+IWINDOW(M1))GO TO 54
                   IF(KOUNT.NE.0)GO TO 53
                   ISWTCH1=1
                   CALL PRT1
                   DO 52 M4=1,21
                   CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1
          1              )
       52          CONTINUE
       53          CALL PRT2(M1,ITM,N2,IREC(ISTO),MRECNT,1)
       54          IF(IREC(ISTO).GE.IOVALUE(M1)-IWINDOW(M1))GO TO 58
                   IF(KOUNT.NE.0)GO TO 56
                   ISWTCH1=1
                   CALL PRT1
                   DO 55 M4=1,21
                   CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1
          1              )
       55          CONTINUE
       56          CALL PRT2(M1,ITM,N2,IREC(ISTO),MRECNT,0)
       58       CONTINUE
                GO TO 69
       59       DO 68 M2=1,18
                   IF(ISWTCH5.EQ.0.AND.M2.LT.3)GO TO 68
                   N2=M2+20*(M1-5)
                   ISTO=LIST(M1-2)+(M2-1)*LISTB(M1-2)
                   IF(IDOT(N2).GE.IBOUND(M1))GO TO 60
                   IBOUND(M1)=IDOT(N2)
                   LFRAME(M1)=ITM
                   LSAMPLE(M1)=ISTO
       60          IF(IDOT(N2).LE.IBOUND(M1+21))GO TO 61
                   IBOUND(M1+21)=IDOT(N2)
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
                  LFRAME(M1+21)=ITM
                  LSAMPLE(M1+21)=ISTO
   61             IF(IDOT(N2).LE.IOVALUE(M1)+IWINDOW(M1))GO TO 64
                  IF(KOUNT.NE.0)GO TO 63
                  ISWTCH1=1
                  CALL PRT1
                  DO 62 M4=1,21
                  CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1
     1            )
   62             CONTINUE
   63             CALL PRT2(M1,ITM,ISTO,IDOT(N2),MRECNT,1)
   64             IF(IDOT(N2).GE.IOVALUE(M1)-IWINDOW(M1))GO TO 68
                  IF(KOUNT.NE.0)GO TO 66
                  ISWTCH1=1
                  CALL PRT1
                  DO 65 M4=1,21
                  CALL PRT2(M4,ITEMPTM,LIST(M4),IOVALUE(M4),MRECNT,-1
     1            )
   65             CONTINUE
   66             CALL PRT2(M1,ITM,ISTO,IDOT(N2),MRECNT,0)
   68           CONTINUE
   69         CONTINUE
              N=0
              DO 70 M=192,228,12
                N=N+1
                DO 70 M1=1,2
                  LASTVAL(N+4*(M1-1))=IREC(M+M1+INDEX)
   70         CONTINUE
              ISWTCH5=1
           GO TO 91
   86      IF(INFLGHT.NE.64)CALL PRT1
   87      ISWTCH5=0
   90 CONTINUE
      GO TO 100
  91    IERRCNT=IERRCNT+2
      ISWTCH5=0
        GO TO 100
  92    ISHTCNT=ISHTCNT+2
        ISWTCH5=0
        GO TO 100
  93    ISWTCH4=1
 100  CONTINUE
      GO TO 20
C
C ANY INCOMPLETE TYPE3 DATA? WRITE LAST TAIL RECORD.
C
 900  IF(ISWTCH4.EQ.0)GO TO 29
      ISWICH3=1
      CALL PRT3
      WRITE(6,1010)
1010  FORMAT(32H1PRINTOUT FOR COMPB2 STATISTICS:    )
      WRITE(6,1050)KNTR1
1050  FORMAT(41H01.  THE NUMBER OF TYPE1 RECORDS WRITTEN:     ,I7)
      WRITE(6,1060)KTNTR2
1060  FORMAT(41H02.  THE NUMBER OF TYPE2 RECORDS WRITTEN:     ,I7)
      WRITE(6,1070)KNTR3
1070  FORMAT(41H03.  THE NUMBER OF TYPE3 RECORDS WRITTEN:     ,I7)
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
      WRITE(6,1074)ITGDCNT
 1074 FORMAT(28H04.  SECONDS OF GROUND DATA*,I7)
      WRITE(6,1076)KTKNT
 1076 FORMAT(26H05.  TOTAL ACTIVE SAMPLES*,I7)
      WRITE (6,1080) MPEAK
 1080 FORMAT(35H06.  THE NUMBER OF PEAKS STORED*     ,I7)
      WRITE(6,1090)NTKNT
 1090 FORMAT(35H07.  THE TOTAL NUMBER OF SECONDS*     ,I7)
      STOP
      END
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
      SUBROUTINE PRT1
      DIMENSION IP1(16),IRECO1(163),ITBL1(103),INDD(12),INDW(19)
      COMMON INDEX, IREC(486)
      COMMON /CNTR/KOUNT, MPEAK, NKOUNT, KNTR1, KNTR2,
     #    KNTR3
      COMMON /CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON /TCNTR/ KTKNT,NTKNT,KTNTR2
      COMMON /TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON /CH3/ IAC,NRECNT
      COMMON /CH4/ ITM
      COMMON/C7/ IRECJ2(96),IDIR(21),IOVALUE(21)
      COMMON /SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      COMMON /THRES/ IWINDOW(21)
      DATA ITBL1/4,-24,52,12*8,4,6*8,12,7*8,4,7*8,4,7*8,4,7*8,4,7*8,4,
     #       7*8,4,7*8,4,7*8,4,7*8,4,7*8,4/
      DATA INDD/48,88,95,96,136,143,144,184,191,192,232,239/
      DATA INDW/3,4,5,6,2,1,7,8,11,12,9,10,13,16,14,15,18,19,17/
C
C*************SWTCH1 DEFINITION***************************************
C     ISWTCH1=0 WHEN IT HAS GONE THROUGH PREVIOUS CALDATA
C     ISWTCH1=1 WHEN IT HAS FOUND THE FIRST GOOD SEC. OF DATA--
C              PRINT TYPE 1 RECORD.
C     ISWTCH1=2 WHEN IT IS TO PROCESS CALDATA THIS
C              ALLOWS FOR INITIALIZATION INTO ARRAY.
C********************************************************************
      IF(ISWTCH1.EQ.1) GO TO 135
      IF(ISWTCH1.EQ.2) INCRST1=40
      INCR=INCRST1-1
C
C   ************** CALIBRATION INFORMATION *****************
C
      DO 100 M=1,17
         INCR=INCR+1
         JINCR=INCR+1
         IF(MOD(JINCR,8).NE.0)GO TO 30
         IRECO1(INCR)=0
         INCR=INCR+1
   30    GO TO (40,40,40,40,40,40,40,40,40,40,50,50,60,60,70,80,90),M
   40    IRECO1(INCR)=IREC(M+INDEX)
         GO TO 100
   50    IRECO1(INCR)=IREC(M+4+INDEX)
         GO TO 100
   60    IRECO1(INCR)=IREC(M+10+INDEX)
         GO TO 100
   70    IRECO1(INCR)=IREC(40+INDEX)
         GO TO 100
   80    IRECO1(INCR)=IREC(47+INDEX)
         GO TO 100
   90    IRECO1(INCR)=IREC(208+INDEX)
  100 CONTINUE
      ISWTCH1=0
      IF(INCR.GE.97)ISWTCH1=2
      INCRST1=INCR
      RETURN
  135 INCR=INCRST1-1
C
C   ************* CREATE HEADING OF THE TYPE3 RECORD *****************
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
C
      ISWTCH3=0
      CALL PRT3
      ISWTCH3=1
      DO 137 M=INCR,103
         IRECO1(M)=0
  137 CONTINUE
      IRECO1(1)=1
      IRECO1(2)=IAC
      IRECO1(3)=0
C
C **************** DOCUMENTARY DATA ********************
C
      DO 120 M=1,12
         IRECO1(M+3)=IREC(INDD(M)+INDEX)
  120 CONTINUE
      IRECO1(16)=0
C
C ******************** WINDOWS ***************************
C ****NY=51 C130****
C
      M1=17
      DO 124 M=1,19
         IF(MOD(M1,8).NE.0)GO TO 123
         IRECO1(M1)=0
         M1=M1+1
  123    IRECO1(M1)=IWINDOW(INDW(M))
         M1=M1+1
  124 CONTINUE
      DO 125 M=38,39
         IRECO1(M)=0
  125 CONTINUE
      IK=IJ=1
      DO 130 M=1,16
         IP1(M)=0
  130 CONTINUE
      CALL PACK(IRECO1,103,IP1,IJ,IK,ITBL1)
      ISWTCH1=2
      KNTR1=KNTR1+1
C     WRITE(7)(IP1(J),J=1,16)
      RETURN
      END
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
      SUBROUTINE PRT2(IDENT,NFRAME,NTIME,JVALUE,KRECPN,IW)
      DIMENSION IPACK(96),IP2(16)
      COMMON/C7/IRECO2(96),IDIR(21),IOVALUE(21)
      COMMON/C8/IBOUND(42),LFRAME(42),LSAMPLE(42)
      COMMON/CNTR/KOUNT,MPEAK,NKOUNT,KNTR1,KNTR2,KNTR3
      DATA IPACK/4,3*12,23*(8,16,8,8)/
      IP=IB=1
      IF(IW.EQ.-1)GO TO 200
      IF(IW.EQ.IDIR(IDENT))GO TO 200
C
C STORAGE OF PEAKS IN THE TYPE2 RECORDS
C  IDENTIFICATION
C  SECOND OF FLIGHT
C  PARAMETER POSITION
C  PEAK VALUE
C
      IRECO2(4*(KRECPN)+1)=IDENT
      IRECO2(4*(KRECPN)+2)=LFRAME(IDIR(IDENT)*21+IDENT)
      IRECO2(4*(KRECPN)+3)=LSAMPLE(IDIR(IDENT)*21+IDENT)
      IRECO2(4*(KRECPN)+4)=IBOUND(IDIR(IDENT)*21+IDENT)
      IDIR(IDENT)=IW
      KOUNT=KOUNT+1
      MPEAK=MPEAK+1
      IF(KRECPN.LT.23)GO TO 300
      IRECO2(1)=2
      IRECO2(2)=IRECO2(3)=IRECO2(4)=0
      DO 50 M=1,16
         IP2(M)=0
   50 CONTINUE
      CALL PACK(IRECO2,96,IP2,IP,IB,IPACK)
C     WRITE(7)(IP2(I),I=1,16)
      KNTR2=KNTR2+1
      KRECPN=0
  300 KRECPN=KRECPN+1
C
C STORAGE OF WINDOW CROSSINGS IN THE TYPE2 RECORDS
C  IDENTIFICATION
C  SECOND OF FLIGHT
C  PARAMETER POSITION
C  WINDOW CROSSING VALUE(NEW MEAN VALUE)
C
  200 IRECO2(4*(KRECPN)+1)=IDENT
      IRECO2(4*(KRECPN)+2)=NFRAME
      IRECO2(4*(KRECPN)+3)=NTIME
      IRECO2(4*(KRECPN)+4)=JVALUE
      IOVALUE(IDENT)=JVALUE
      LFRAME(IDENT)=LFRAME(IDENT+21)=NFRAME
      LSAMPLE(IDENT)=LSAMPLE(IDENT+21)=NTIME
      IBOUND(IDENT)=IBOUND(IDENT+21)=JVALUE
      KOUNT=KOUNT+1
      IF(KRECPN.LT.23)GO TO 100
      IP=IB=1
      IRECO2(1)=2
      IRECO2(2)=IRECO2(3)=IRECO2(4)=0
      DO 150 M=1,16
         IP2(M)=0
  150 CONTINUE
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
                  CALL PACK(IREC02,96,IP2,IP,IB,IPACK)
          C       WRITE(7)(IP2(I),I=1,16)
   60             KNTR2=KNTR2+1
                  KRECPN=0
          100     KRECPN=KRECPN+1
                  RETURN
                  END
```

TABLE A.4 (Continued)

PROGRAM COMPB2


```
      SUBROUTINE PRT3
      DIMENSION IP3(17),IRECO3(17),ITBL3(17),INDD(12)
      COMMON INDEX,IREC(486)
      COMMON/C7/IRECO2(96),IDIR(21),IOVALUE(21)
      COMMON /CM3/ IAC,MRECNT
      COMMON /CM4/ ITM
      COMMON /CNTR/ KOUNT,MPEAK,MKOUNT,KNTR1,KNTR2,KNTR3
      COMMON /CNTR2/ IKOUNT,IGRDCNT,IERRCNT,ISHTCNT
      COMMON /TCNTR/ KTKNT,NTKNT,KTNTR2
      COMMON /TCNTR2/ ITKNT,ITGDCNT,ITERCNT,ITSTCNT
      COMMON /SWTCH/ ISWTCH1,ISWTCH2,ISWTCH3,IEND
      DATA ITBL3/4,-24,52,12*8,4,60/
      DATA INDD/48,88,95,96,136,143,144,184,191,192,232,239/
C
C ********  END OF FLIGHT  (TAIL:  *************************
C *********  SWTCH3 DEFINITION *************
C       ISWTCH3=0     WHEN INITIAL ENTRY STORE DOC.DATA
C       ISWTCH3=1     WHEN ITS TO SAVE TIME SEC. AND WRITE
C                     TYPE3 RECORD
C ***********************************************************
C
      IF(ISWTCH3.EQ.1)GO TO 11
      IRECO3(1)=3
      IRECO3(2)=IAC
      IRECO3(3)=IAC
C
C *************** DOCUMENTARY DATA ************************
C
      DO 10 M=1,12
        IRECO3(M+2)=IREC(INDD(M)+INDEX)
   10 CONTINUE
      ISWTCH3=1
      RETURN
   11 IRECO3(16)=0
      IRECO3(17)=ITM
      IEND=1
      ISWTCH3=0
      IF(KOUNT.EQ.0)GO TO 30
C
C CLEAR OUT THE REMAINING TYPE2 RECORDS BEFORE WRITING
C OUT THE TAIL RECORD.
C
      IF(MRECNT.EQ.1)GO TO 18
      IF(MRECNT.EQ.23)GO TO 15
      DO 12 K7=MRECNT,23
        IRECO2(4*(K7)+1)=IRECO2(4*(K7)+2)=0
        IRECO2(4*(K7)+3)=IRECO2(4*(K7)+4)=0
   12 CONTINUE
      MRECNT=23
   15 CALL PRT2(1,0,0,0,MRECNT,-1)
      KOUNT=KOUNT-1
   18 KNTR3=KNTR3+1
      IJ=1
      IK=1
      DO 20 M=1,16
        IF3(M)=0
   20 CONTINUE
```

TABLE A.4 (Concluded)

PROGRAM COMPB2


```
      CALL PACK(IRECO3,17,IP3,IJ,IK,ITBL3)
C     WRITE(7)(IP3(J),J=1,16)
      WRITE(6,1010)KNTR1
 1010 FORMAT(8H0FLIGHT#,I3,11H STATISTICS)
      WRITE(6,1020)KNTR2
 1020 FORMAT(41H 1.  THE NUMBER OF TYPE2 RECORDS PRINTED=,I7)
      WRITE(6,1060)IGRDCNT
 1060 FORMAT(28H 2.  SECONDS OF GROUND DATA=,I7)
      WRITE(6,1090)KOUNT
 1090 FORMAT(28H 3.  SAMPLES OF ACTIVE DATA=,I7)
      WRITE(6,1110)ITM
 1110 FORMAT(29H 4.  TOTAL NUMBER OF SECONDS=,I7)
   30 MKOUNT=MKOUNT-ITM
      KTNTR2=KNTR2+KTNTR2
      ITERCNT=IERRCNT+ITERCNT
      ITSTCNT=ISHTCNT+ITSTCNT
      ITGDCNT=IGRDCNT+ITGDCNT
      ITKNT=IKOUNT+ITKNT
      KTKNT=KOUNT+KTKNT
      NTKNT=ITM+NTKNT
      KNTR2=IERRCNT=ISHTCNT=IGRDCNT=0
      IKOUNT=KOUNT=0
      RETURN
      END
```

TABLE A.5

SUBROUTINE PACK/UNPACK

```
      SUBROUTINE UNPACK(UP,NUP,P,J,NB,ITBL)
      INTEGER P,UP
      DIMENSION UP(1),P(1),ITBL(1),MSK(60),MSKB(10)
      DATA IM/0/
      DATA MSKB/0,55B,5555B,555555B,55555555B,5555555555B,555555555555B,
     *55555555555555B,5555555555555555B,555555555555555555B/
      IF(IM.NE.0) GO TO 10
      MSK(1)=COMPL(0)
      DO 5 I=2,60
    5    MSK(I)=COMPL(MASK(I-1))
      IM=1
   10 CONTINUE
      DO 500 I=1,NUP
      JJ=ITBL(I)
      IF(IABS(JJ).GT.(61-NB)) GO TO 50
      IF(JJ) 20,500,15
   15 UP(I)=AND(SHIFT(P(J),NB+JJ-1),MSK(61-JJ))
      NB=NB+JJ
      GO TO 30
   20 JJ=-JJ
      UP(I)=SHIFT(AND(SHIFT(P(J),NB+JJ-1),MSK(61-JJ)),60-JJ)
      NB=NB+JJ
   25 IF(MOD(JJ,6).NE.0) GO TO 30
      LL=11-JJ/6
      UP(I)=OR(UP(I),MSKB(LL))
   30 IF(NB.LT.61) GO TO 500
      NB=1
      J=J+1
      GO TO 500
   50 IF(JJ.LT.0) GO TO 60
      KK=61-NB
      LL=JJ-KK
      UP(I)=SHIFT(AND(MSK(NB),P(J)),LL)
      J=J+1
      UP(I)=OR(UP(I),AND(SHIFT(P(J),LL),MSK(61-LL)))
      NB=1+LL
      GO TO 30
   60 JJ=-JJ
      UP(I)=SHIFT(AND(MSK(NB),P(J)),NB-1)
      J=J+1
      KK=JJ-61+NB
      UP(I)=OR(UP(I),SHIFT(AND(SHIFT(P(J),KK),MSK(61-KK)),60-JJ))
      NB=1+KK
      GO TO 25
  500 CONTINUE
      RETURN
      END
```

TABLE A.5 (Concluded)

SUBROUTINES PACK/UNPACK

```
      SUBROUTINE PACK(UP,NUP,P,J,NB,ITBL)
C     UP IS UNPACKED ARAY
C     NUP IS NUMBER OF WORDS FROM UP TO BE PACKED
C     P IS PACKED ARRAY
C     J IS FIRST AVAILABLE WORD IN P ARRAY
C     NB IS FIRST AVAILABLE BIT IN JTH WORD OF P ARRAY(NB GE 1 OR LE 60)
C     ITBL IS TABLE OF LENGTH NUP CONTAINING NUMBER OF BITS FROM
C           EACH VALUE IN UP TO PACKED IN P
      INTEGER P,UP
      DIMENSION UP(1),P(1),ITBL(1),MSK(60)
      DATA IM/0/
      IF(IM.NE.0) GO TO 10
      MSK(1)=COMPL(0)
      DO 5 I=2,60
    5    MSK(I)=COMPL(MASK(I-1))
      IM=1
   10 CONTINUE
      DO 500 I=1,NUP
      JJ=ITBL(I)
      IF(IABS(JJ).GT.(61-NB)) GO TO 50
      IF(JJ) 20,500,15
   15 P(J)=OR(P(J),SHIFT(AND(UP(I),MSK(61-JJ)),61-NB-JJ))
      NB=NB+JJ
      GO TO 30
   20 JJ=-JJ
      P(J)=OR(P(J),SHIFT(AND(SHIFT(UP(I),JJ),MSK(61-JJ)),61-NB-JJ))
      NB=NB+JJ
   30 IF(NB.LT.61) GO TO 500
      NB=1
      J=J+1
      GO TO 500
   50 IF(JJ.LT.0) GO TO 60
      KK=121-JJ-NB
      P(J)=OR(P(J),AND(SHIFT(UP(I),KK),MSK(NB)))
      J=J+1
      P(J)=OR(P(J),SHIFT(AND(UP(I),MSK(KK+1)),KK))
      NB=61-KK
      GO TO 30
   60 JJ=-JJ
      P(J)=OR(P(J),AND(MSK(NB),SHIFT(UP(I),61-NB)))
      J=J+1
      KK=121-JJ-NB
      P(J)=OR(P(J),SHIFT(AND(MSK(KK+1),SHIFT(UP(I),JJ)),KK))
      NB=61-KK
      GO TO 30
  500 CONTINUE
      RETURN
      END
```

TABLE A.6

PROGRAMS COMPA1 AND COMPA2 DEFINITION OF VARIABLES

IAC - aircraft title

IERRCNT - counter incremented when a parity error has been found

IGRDCNT - counter incremented if second of data was recorded on the ground

IKOUNT - counter incremented when active data is found

INDEX - used to determine first or second half of frame

INFLIGHT - flag for in-flight

IOVALT - old value of the altitude

IOVAS - old value of the air speed

IOVE1 - old value of the events of 1-4

IOVE5 - old value of the events of 5-9

IOVNY - old value of the NY parameter

IOVNY - old value of the NZ parameter

IOVQ - old value of the Q parameter

IOVR - old value of the R parameter

IQDOT - array used for the integrated q value

IRDOT - array used for the integrated r value

IREC - array used for 2 seconds of unpacked data

ISHTCNT - counter incremented when a short record has been found

ISWTCH1 - switch used in PRT1 routine
= 0 when the routine has gone through previous calibration data
= 1 when the routine is to write the record
= 2 when the routine is to initialize INCR variable

ISWTCH2 - switch used in PRT3 routine
= 0 when the routine is to store time and data.
= 1 when the routine is to store values of $\dot{p}$, $\dot{q}$, and $\dot{r}$ and write the record

ISWTCH3 - switch used in PRT4 routine
= 0 when the routine is to store the documentary data
= 1 when the routine is to write the record

ISWTCH4 - switch used in the MAIN routine
= 0 when the routine is to skip over seconds read in
= 1 when processing is finished upon reading a tape mark

ITALT - altitude threshold

ITAS - airspeed threshold

TABLE A.6 (Continued)

PROGRAMS COMPA1 AND COMPA2 DEFINITION OF VARIABLES

ITBL - unpacking scheme for RT record

ITERCNT - total parity errors counter

ITGDCNT - total seconds of ground data

ITKNT - total active data counter

ITM - time parameter

ITNY - N_y threshold

ITNZ - N_z threshold

ITQ - Q threshold

ITR - R threshold

ITSTCNT - total short records counter

JBIT - flag turned on during a built in cycle test

JCAL - flag turned on during a calibration cycle

JPRE - flag indicating PRE bit is on

KNTR1 - counter used in counting the type 1 records

KNTR2 - counter used in counting the type 2 records

KNTR3 - counter used in counting the type 3 records

KNOUNT - counter used in counting the active data found

KTKNT - total active data found

KTNTR2 - total type 2 records written

LAST - flag signifying if the previous second was active or inactive

LKOUNT - counter used in counting the inactive data found

LTKNT - total inactive seconds found

NGR - switch indicating whether to process ground data or not

KNOUNT - counter used in counting number of seconds read in

NTBLK - array into which data is read from tape

NTKNT - total number of seconds read in

Subroutine ACTDATA

INCR - variable used to index the LASTQ and LASTR arrays

LASTQ - array used to store the last four pitch values

LASTR - array used to store the last four yaw values

TABLE A.6 (Concluded)

PROGRAMS COMPA1 AND COMPA2 DEFINITION OF VARIABLES

Subroutine PRT1

INCR - variable used to index the IRECO1 array during different calibration periods

INCRST1 - variable used to save INCR when passed back to the main routine

INDD - array containing the location of the documentary data in IREC

IP1 - array returned from the PACK routine in packed format

IRECO1 - array given to the PACK routine in unpacked format

ITBL1 - packing scheme for type 1 record

Subroutine PRT2

IP2 - array returned from the PACK routine in packed format

IRECO2 array given to the PACK routine in unpacked format

ITBL2 - packing scheme for type 2 record

Subroutine PRT3

IP3 - array returned from the PACK routine in packed format

IRECO3 - array given to the PACK routine in unpacked format

ITBL3 - packing scheme for type 3 record

TABLE A.7

PROGRAM COMPB1 AND COMPB2 DEFINITION OF VARIABLES

IAC - aircraft title

IBOUND - array containing minimums of each parameter in the first half, and maximums in second half

IDIR - previous direction of parameter (1 = up, 0 = down)

IDOT - array of $\dot{p}$, $\dot{q}$ and $\dot{r}$ values

IERRCNT - number of seconds containing parity errors

IGRDCNT - number of seconds of ground data

IKOUNT - number of BIT/PRE/CAL seconds

INDEX - used to determine first or second half of frame

INFLGHT - flag for in-flight

IOVALUE - array containing last window crossing

IREC - unpacked input data

ISHTCNT - number of short record seconds

ISWTCH1 - used in PRT1 routine (same as in program COMPA1)

ISWTCH3 - used in PRT3 routine (same as in program COMPA1)

ISWTCH4 - used in main routine (same as in program COMPA1)

ITBL - unpacking scheme for RT record

ITERCNT - total seconds with parity errors

ITGDCNT - total seconds of ground data

ITKNT - total of BIT/PRE/CAL seconds

ITSTCNT - total short record seconds

ITM - time into flight

IW - current parameter direction (see IDIR)

IWINDOW - window values

JBIT, JCAL, JPRE - flags for BIT, CAL, PRE seconds, respectively

KNTR1, KNTR2, KNTR3 - counters for type 1, 2 3 records, respectively

KTKNT - total samples of active data

KTNTR2 - total number of type 2 records

LFRAME - frames containing minimums or maximums accessed like IBOUND

LIST - location of parameters in IREC array

LSAMPLE - sample numbers for minimums and maximums assessed like IBOUND

IRECO2 - Type 2 buffer area

MPEAK - number of peaks stored

TABLE A.7 (Concluded)

PROGRAM COMPB1 AND COMPB2 DEFINITION OF VARIABLES

MRECNT - type 2 data pointer

NGR - flag for inflight

NKOUNT - number of seconds

NTBLK - input buffer for packed data

USED ONLY IN C141, C130:

LISTA - number of samples/sec for each parameter

LISTB - displacement factor for each parameter

Subroutine PRT1

INCR - index IRECO1 for calibration

INCRST1 - save area for INCR

INDD - documentary data

INDW - window indexing scheme

IP1 - packed type 1 record

IRECO1 - unpacked type 1 record

ITBL1 - type 1 packing scheme

Subroutine PRT2

IDENT - parameter identification

IPACK - type 2 packing scheme

JVALUE - window crossing

KRECPN - position within type 2 record

IP2 - output buffer for packed type 2 record

NFRAME - frame number of value

NTIME - time of value

Subroutine PRT3

IP3 - packed type 3 data

IRECO3 - unpacked type 3 data

ITBL3 - packing scheme for type 3 records

MAIN PROGRAM

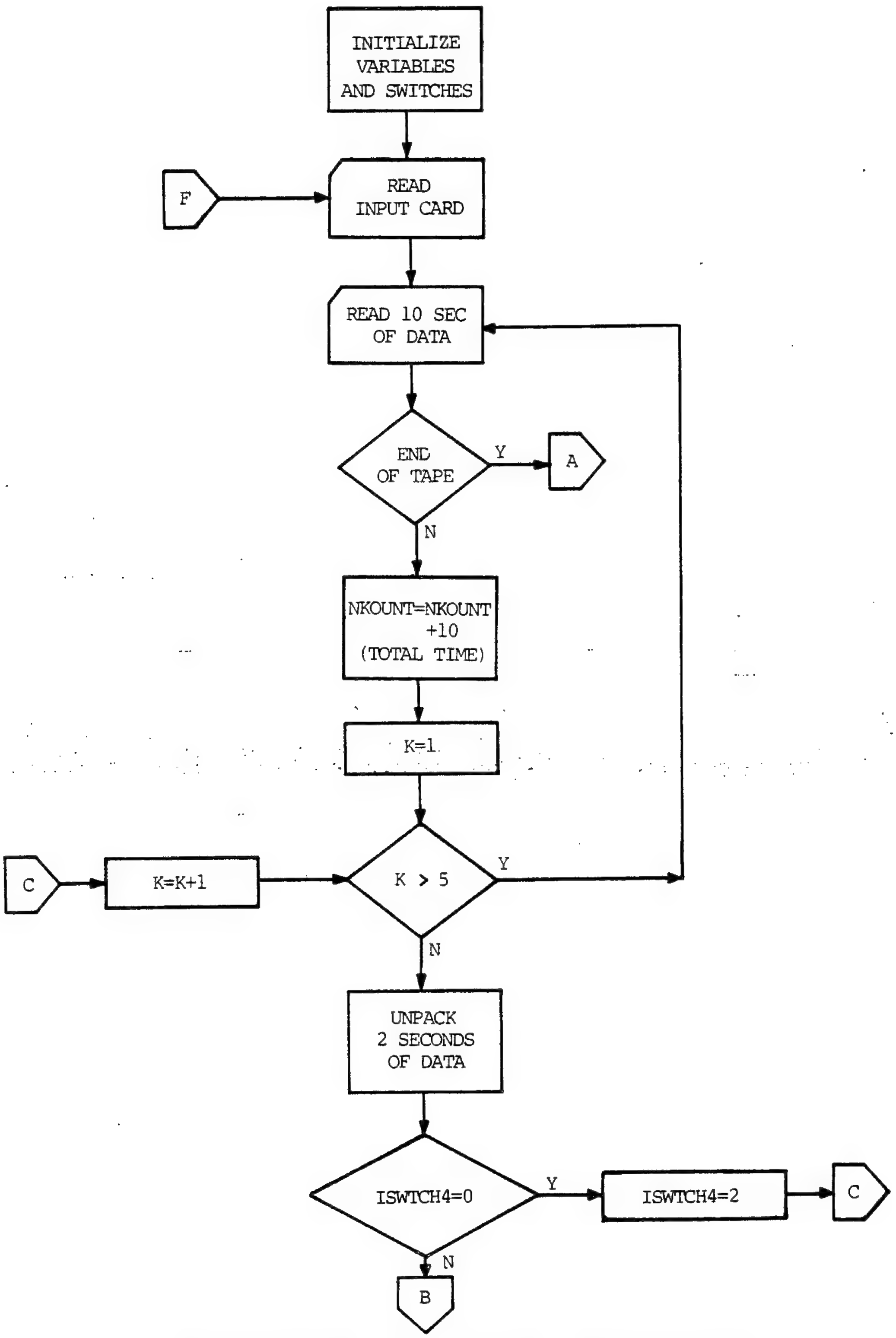


Figure A.1. Flowchart of Method A.

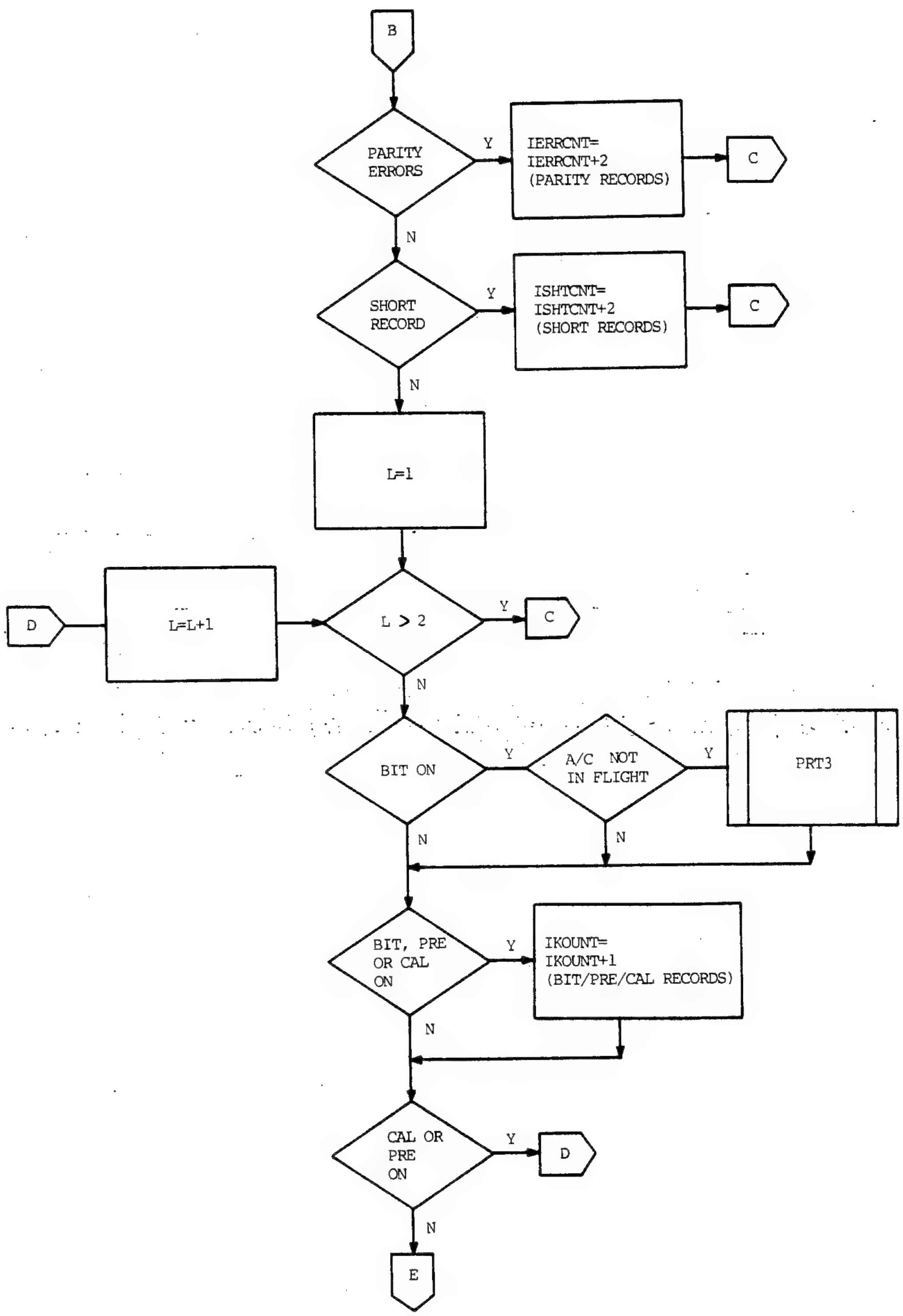


Figure A.1. (Continued)

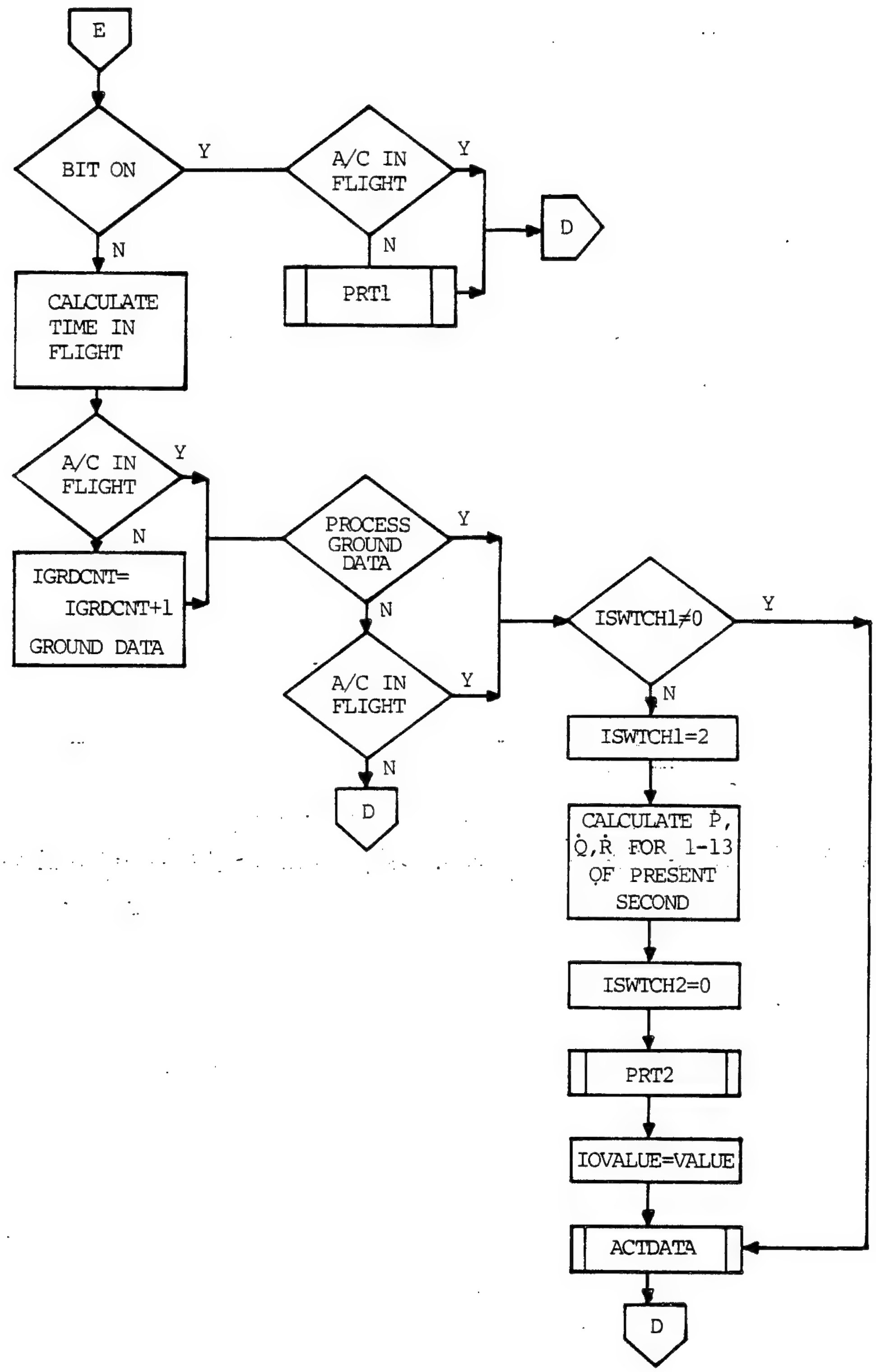


Figure A.1. (Continued)

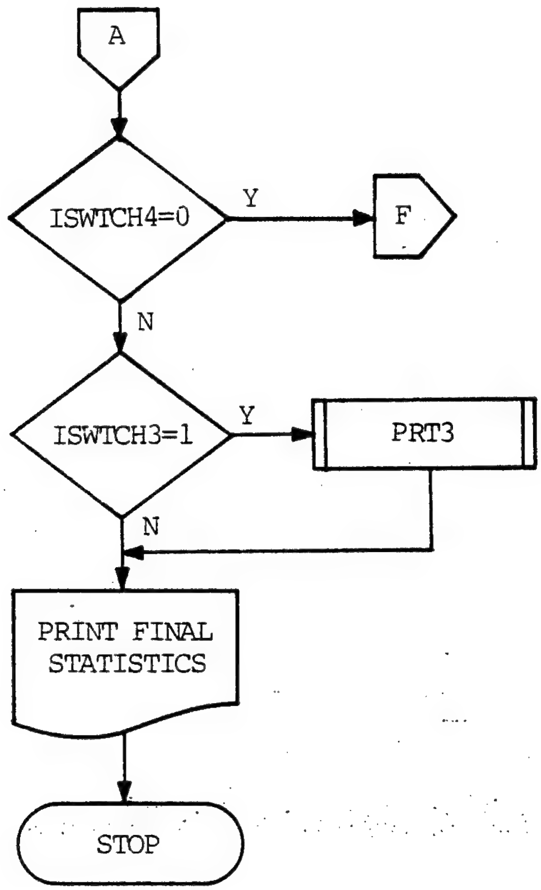


Figure A.1. (Continued)

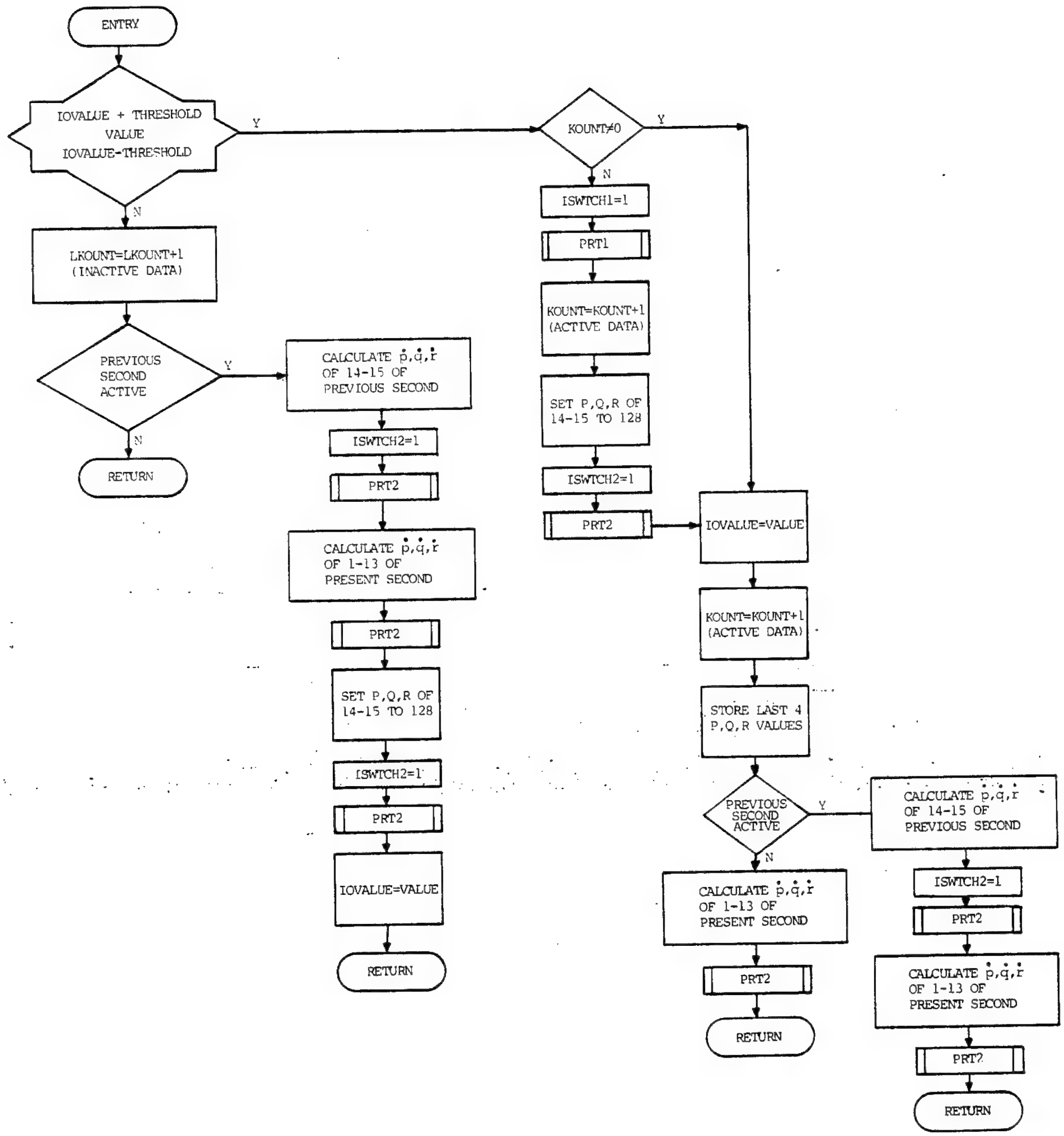


Figure A.1. (Continued)

SUBROUTINE PRT1

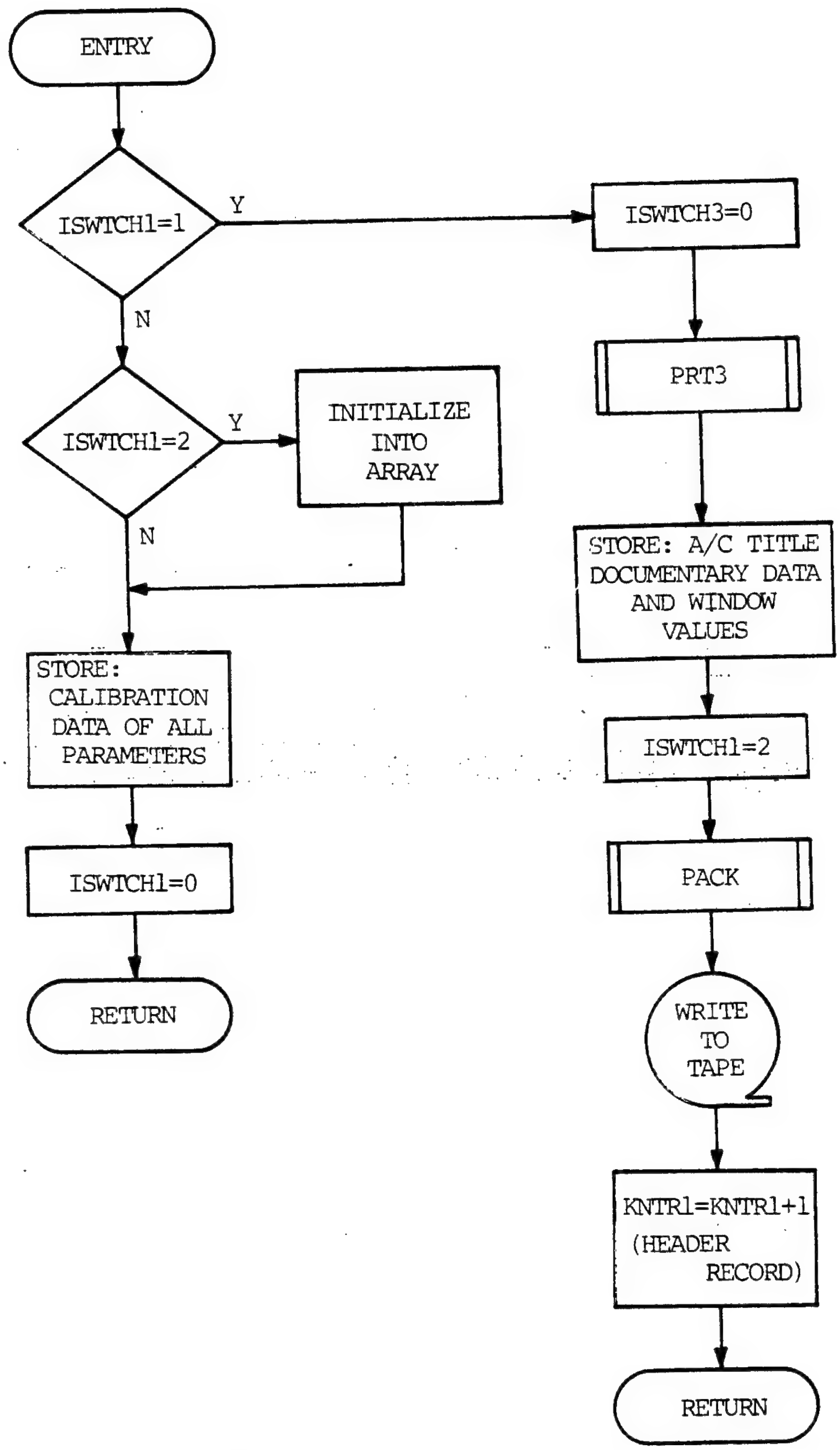


Figure A.1. (Continued)

SUBROUTINE PRT2

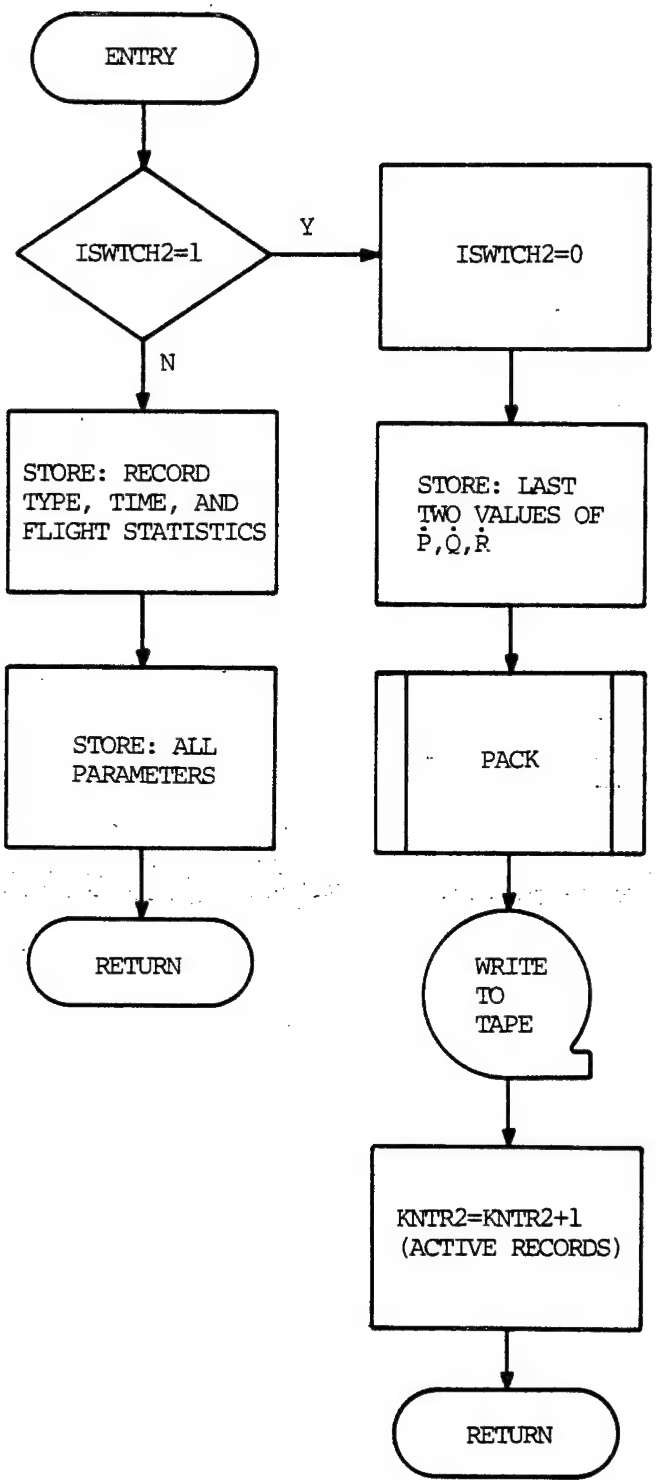


Figure A.1. (Continued).

SUBROUTINE PRT3

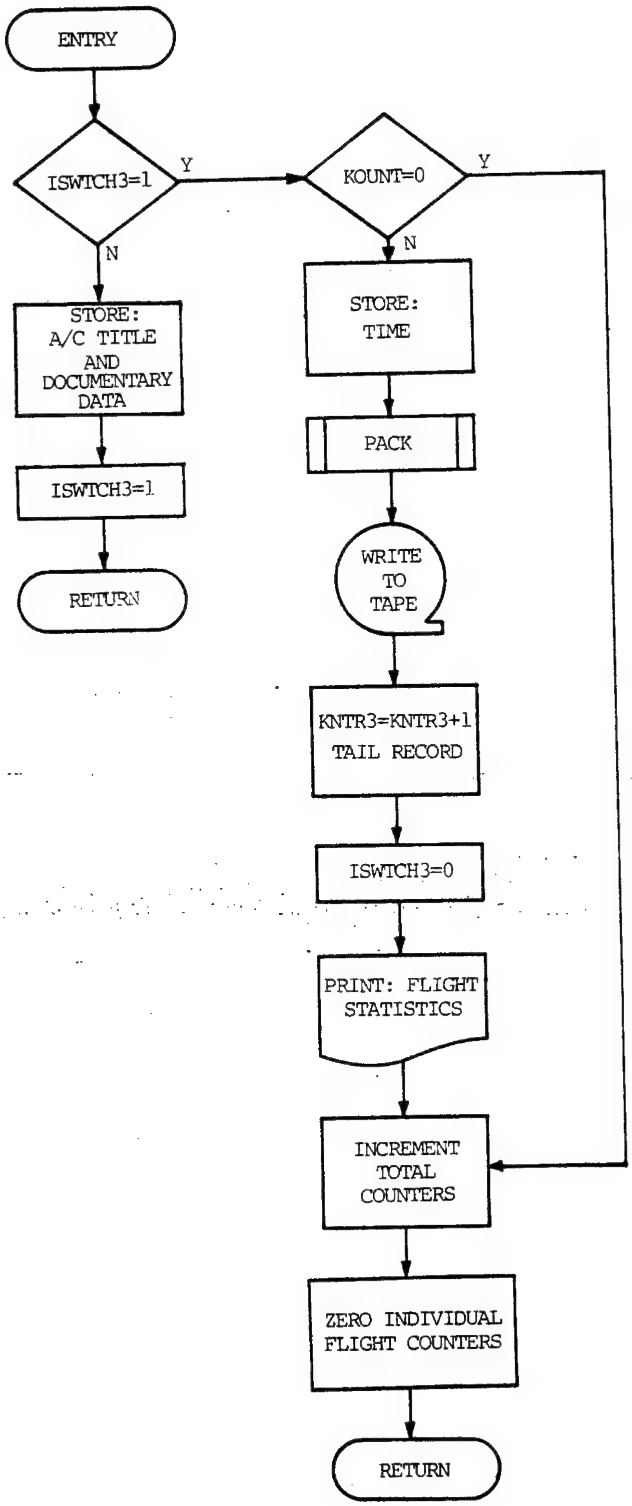


Figure A.1. (Concluded).

MAIN PROGRAM

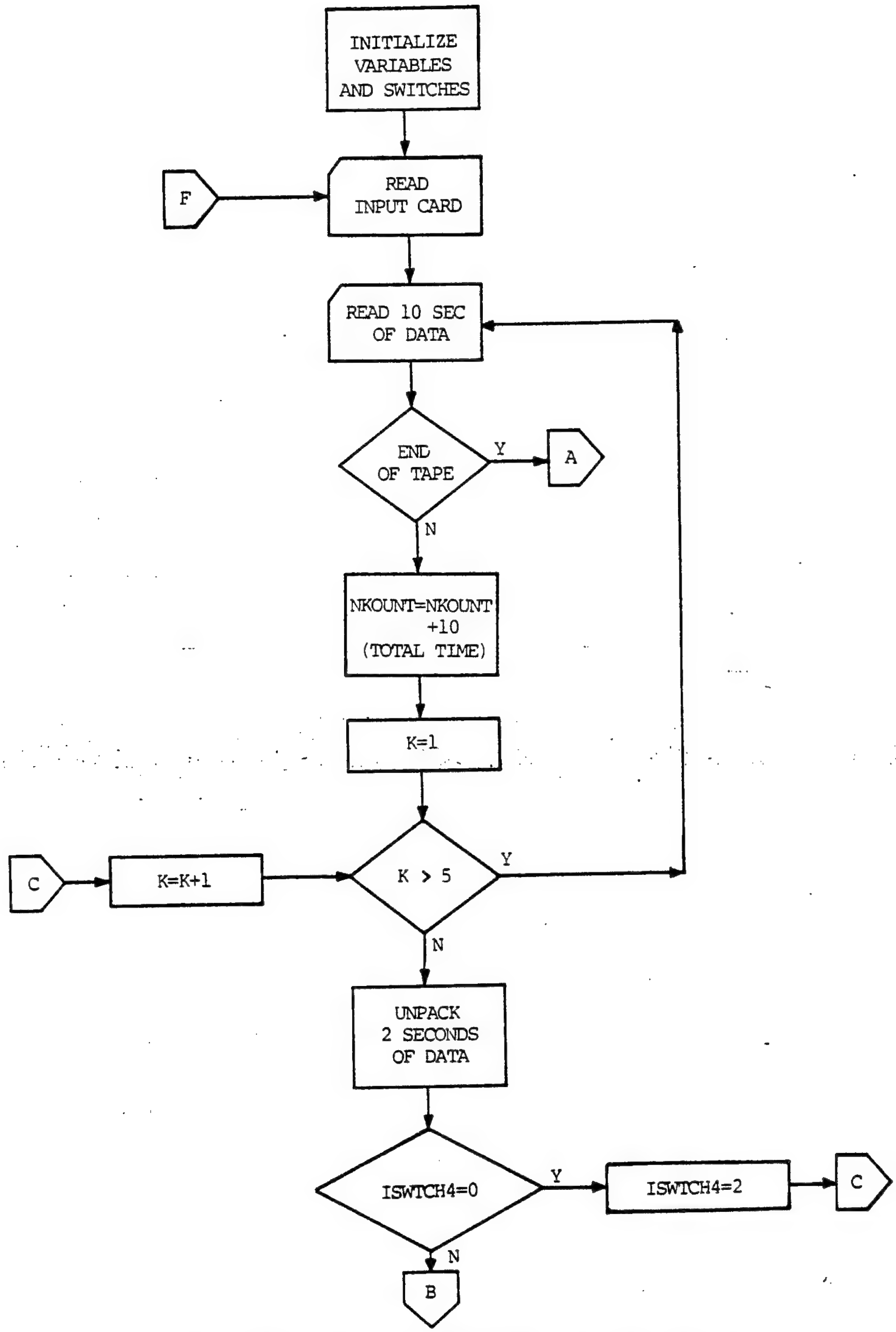


Figure A.2. Flowchart of Method B.

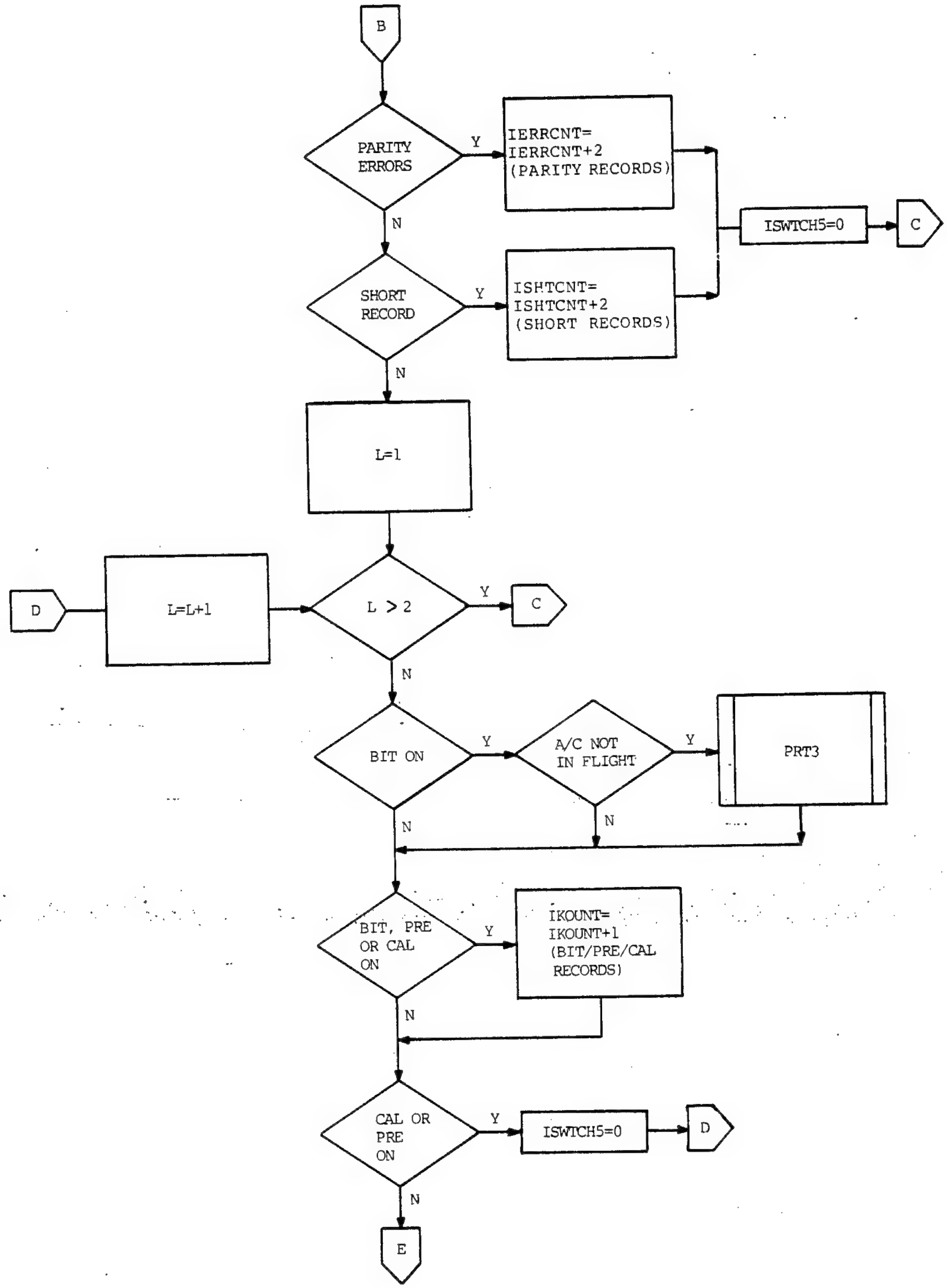


Figure A.2. (Continued).

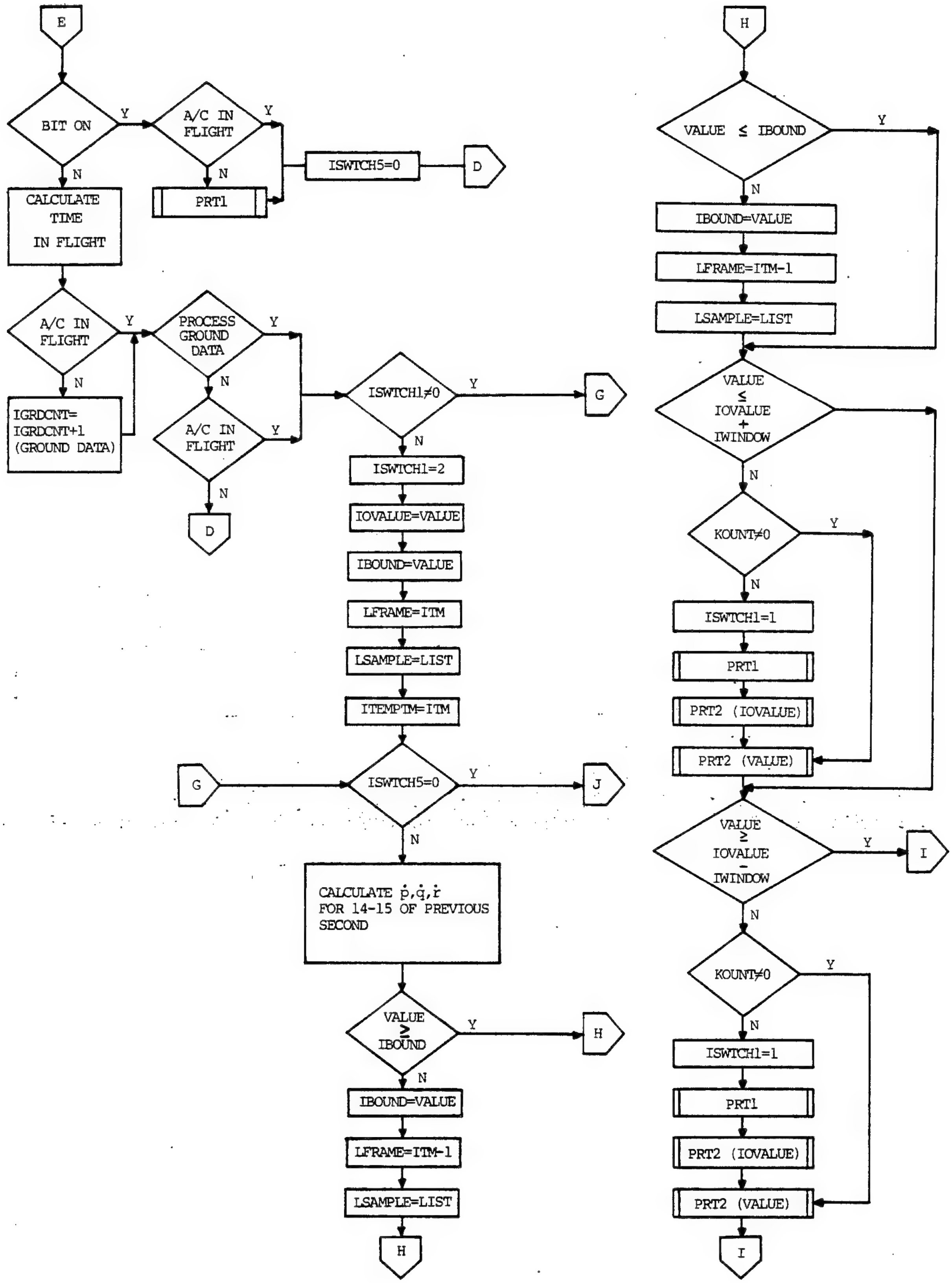


Figure A.2. (Continued).

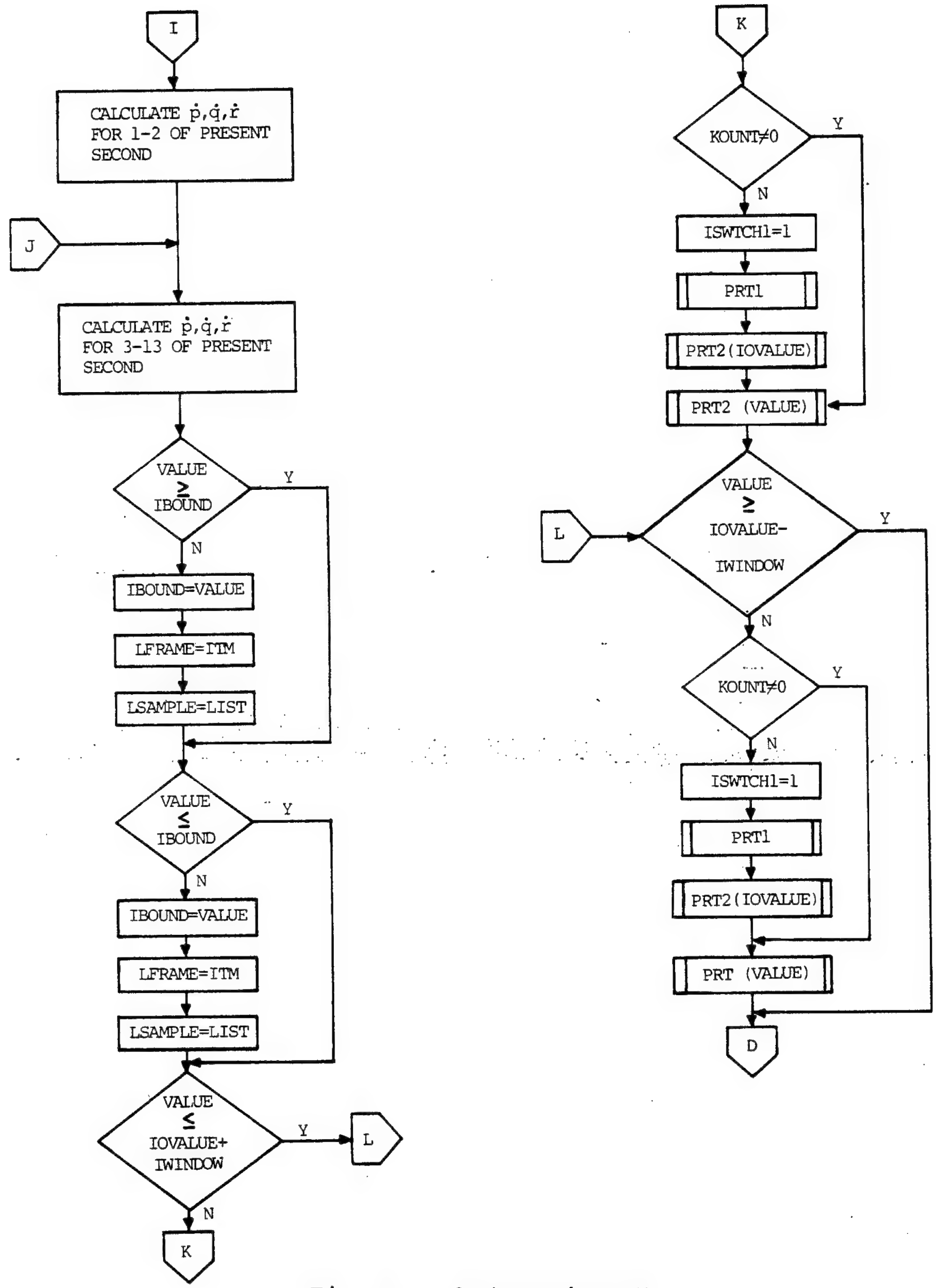


Figure A.2 (Continued).

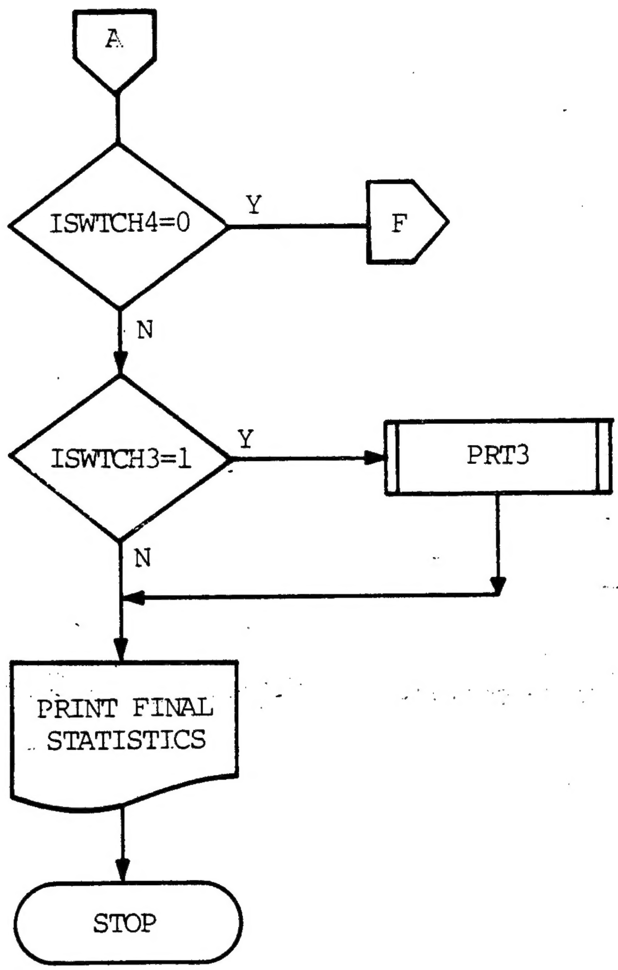


Figure A.2. (Continued).

SUBROUTINE PRT1

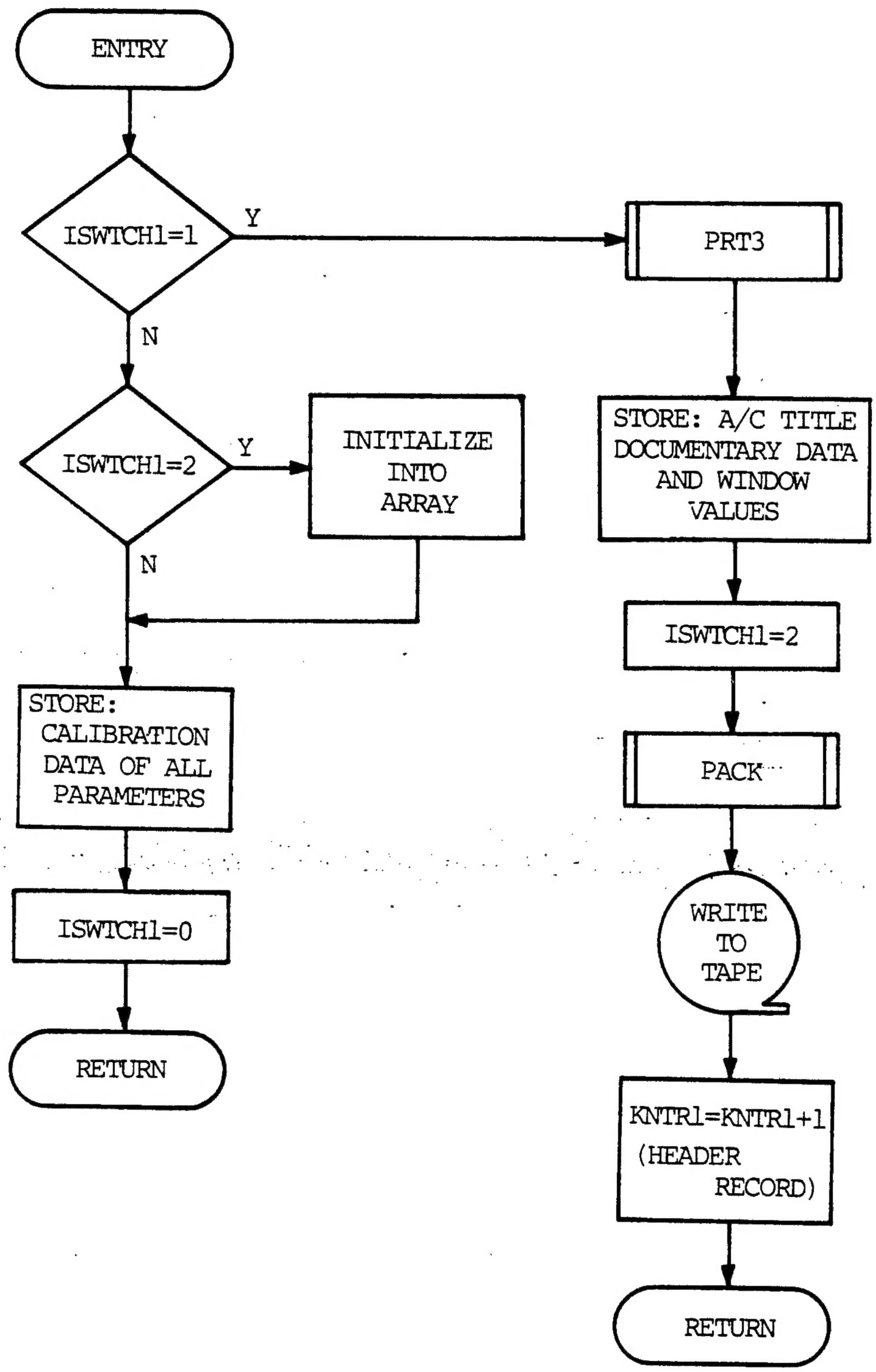


Figure A.2. (Continued).

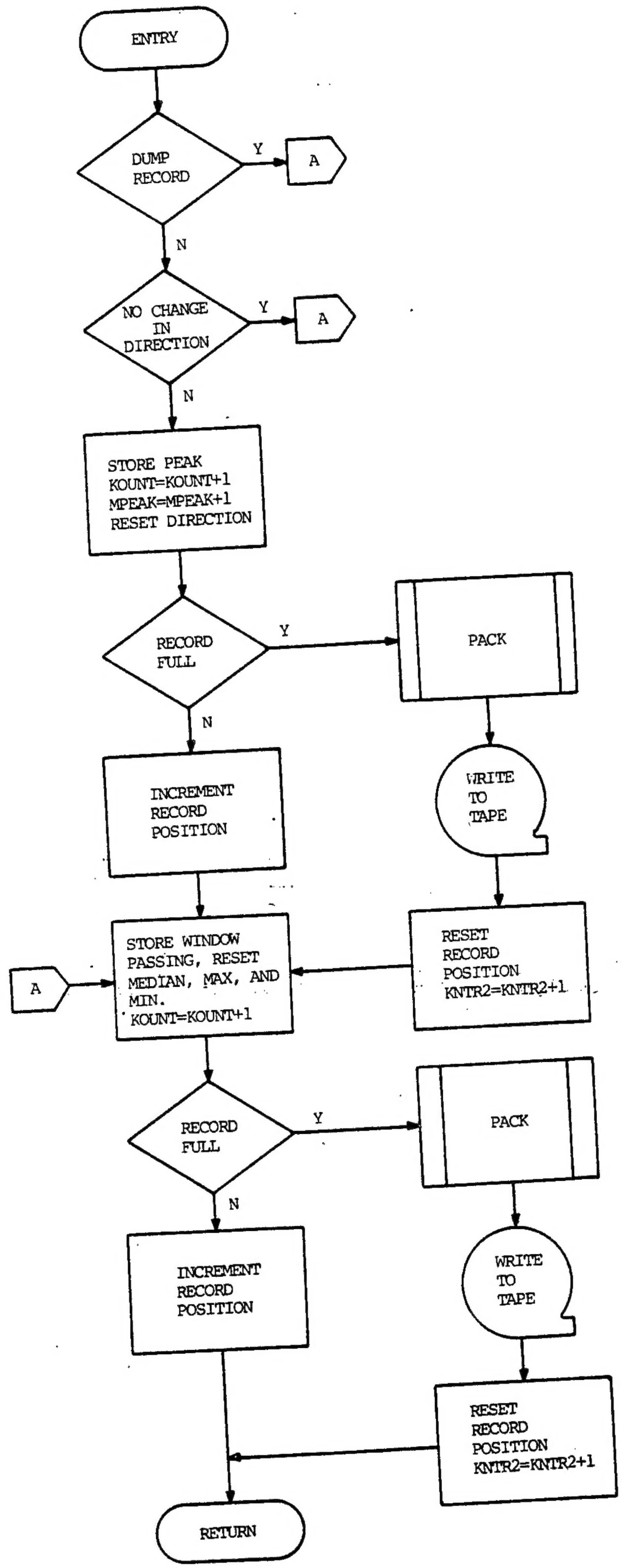


Figure A.2. (Continued).

SUBROUTINE PRT3

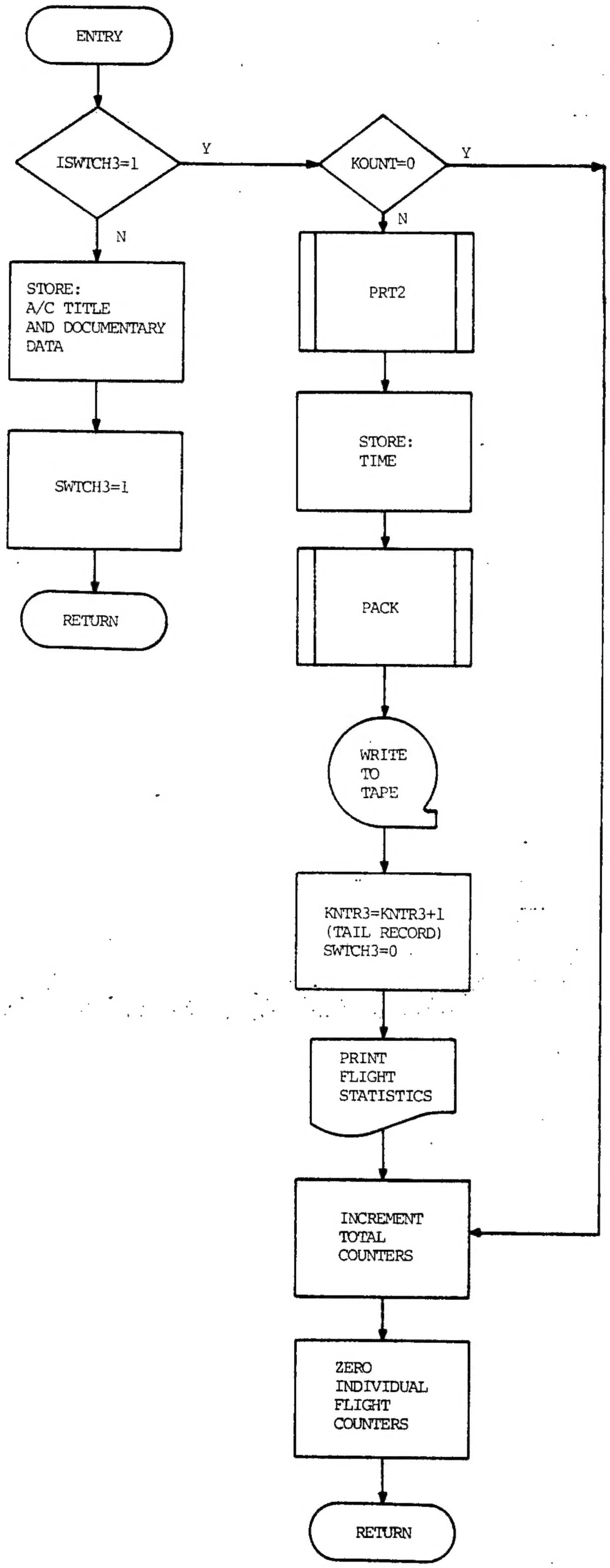


Figure A.2. (Concluded).